



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

**TUGAS AKHIR - TE 141599**

***OPTIMAL CONTROL PERMANENT MAGNET DIRECT  
CURRENT (PMD) MOTOR MENGGUNAKAN PARTICLE  
SWARM OPTIMIZATION (PSO)***

Rodhi Kelvianto  
NRP 2211100195

Dosen Pembimbing  
Prof. Dr. Ir. Imam Robandi, MT  
Ir. Teguh Yuwono

JURUSAN TEKNIK ELEKTRO  
Fakultas Teknologi Industri  
Institut Teknologi Sepuluh Nopember  
Surabaya 2016



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

**FINAL PROJECT-TE 141599**

**OPTIMAL CONTROL PERMANENT MAGNET DIRECT  
CURRENT (PMDC) MOTOR USING PARTICLE SWARM  
OPTIMIZATION (PSO)**

Rodhi Kelvianto  
NRP 2211100195

Advisor  
Prof. Dr. Ir. Imam Robandi, MT  
Ir. Teguh Yuwono

DEPARTMENT OF ELECTRICAL ENGINEERING  
Faculty of Industrial Technology  
Sepuluh Nopember Institute of Technology  
Surabaya 2016

**OPTIMAL CONTROL PERMANENT MAGNET  
DIRECT CURRENT (PMDC) MOTOR  
MENGUNAKAN PARTICLE SWARM  
OPTIMIZATION (PSO)**

**TUGAS AKHIR**

Dijadikan Guna Memenuhi Sebagian Persyaratan  
Untuk Memperoleh Gelar Sarjana Teknik  
Pada

Bidang Studi Teknik Sistem Tenaga

Jurusan Teknik Elektro

Fakultas Teknologi Industri

Institut Teknologi Sepuluh Nopember

Menyetujui :

Dosen Pembimbing I,

Prof. Dr. Ir. Iwan Rohandi, MT.

NIP 19630817194031001

Dosen Pembimbing II,

Ir. Triadi Yudianto

NIP 195008061974121002

SURABAYA  
FEBRUARI 2016

# ***Optimal Control Permanent Magnet Direct Current (PMDC) Motor Menggunakan Particle Swarm Optimization (PSO)***

Rodhi Kelvianto  
2211100195

Dosen Pembimbing I : Prof. Dr. Ir. Imam Robandi, MT  
Dosen Pembimbing II : Ir. Teguh Yuwono

## **Abstrak :**

Sistem tenaga listrik merupakan unit usaha yang faktor teknisnya, termasuk efisiensi dan performansi sistem merupakan hal yang sangat penting untuk diperhatikan. Gangguan pada sistem tenaga listrik terjadi karena adanya perubahan beban. Perubahan beban dapat mempengaruhi nilai parameter-parameter penting, terutama perubahan pada frekuensi yang dapat mengakibatkan hilangnya sinkronisasi pada sistem tenaga listrik. Selain itu, perkembangan teknologi semakin meningkatkan kompleksitas dari sistem perindustrian. Pada tugas akhir ini, dilakukan kontrol *Permanent Magnet Direct Current (PMDC) Motor* yang ditala menggunakan *Particle Swarm Optimization (PSO)*. Peralatan-peralatan industri seperti motor DC sangat membutuhkan suplai tegangan yang sesuai agar kecepatan yang diperoleh sesuai dengan yang dibutuhkan. Motor PMDC berguna untuk memberikan suplai daya aktif saat terjadi perubahan beban sehingga frekuensi dapat terus dijaga serta tegangan pada sistem selalu terjaga pada kondisi stabil. Kontroler yang digunakan pada Tugas Akhir ini adalah kontroler PI. Dengan menggunakan kontrol *Permanent Magnet Direct Current (PMDC) Motor* diharapkan osilasi pada sistem dapat teredam dengan baik. Untuk mendapatkan hasil yang optimal, maka digunakan metode optimisasi menggunakan algoritma *Particle Swarm Optimization*.

***Kata kunci : PMDC Motor, kontroler PI, PSO***

# ***Optimal Control Permanent Magnet Direct Current (PMDC) Motor Using Particle Swarm Optimization (PSO)***

Rodhi Kelvianto  
2211100195

Supervisor I : Prof. Dr. Ir. Imam Robandi, MT  
Supervisor II : Ir. Teguh Yuwono

## **Abstract :**

*Electric power system is a business unit of technical factors, such as the efficiency and performance of the system is very important. Disturbances in the power system caused by load changes. Load changes can affect the value of important parameter, especially the changes in frequency which can lead to loss of synchronization of the power system. In addition, technological developments increase the complexity of industrial systems. This final project is about Optimal Control Permanent Magnet Direct Current (PMDC) Motor Using Particle Swarm Optimization (PSO). Industrial equipment such as DC motor needs supply voltage in order to speed obtained as appropriate required. PMDC Motor gives active power supply when there is a load change so frequency can be maintained as well as the frequency and the voltage in the system is always maintained in a stable condition. This final project uses PI controller. By using the control of Permanent Magnet Direct Current (PMDC) motor is expected oscillations in the system can be damped well. To obtain optimal results, and then used optimization method using Particle Swarm Optimization algorithm.*

**Keywords :** PMDC Motor, PI controller, PSO

## KATA PENGANTAR

Alhamdulillah, segala puji syukur penulis panjatkan kepada Allah yang telah memberikan pertolongan dan kemudahan sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul :

### ***Optimal Control Permanent Magnet Direct Current (PMDC) Motor Menggunakan Particle Swarm Optimization (PSO)***

Tugas Akhir ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik (ST) pada bidang studi teknik sistem tenaga Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember.

Penulis mengucapkan terima kasih kepada semua pihak yang telah memberikan bantuan dan dukungan dalam menyelesaikan Tugas Akhir ini, terutama kepada :

1. Kedua orang tua yang telah memberikan doa, motivasi, bimbingan, dan nasehat dalam pengerjaan Tugas Akhir ini.
2. Bapak Prof. Dr. Ir. Imam Robandi, MT. selaku dosen pembimbing yang telah memberikan bimbingan, kritik, saran, dan motivasi kepada penulis saat menjadi anggota laboratorium PSOC maupun selama menyelesaikan tugas akhir.
3. Bapak Ir. Teguh Yuwono, selaku dosen pembimbing II yang telah memberikan bimbingan, pengarahan, dan koreksi kepada penulis mulai dari analisis data, penyusunan proposal sampai *finishing* dari Tugas Akhir ini.
4. Bapak Dr. Ardyono Priyadi, ST., M.Eng. selaku Kajur Teknik Elektro ITS dan seluruh dosen Jurusan Teknik Elektro ITS, yang telah memberikan bimbingan dan ilmu pengetahuan selama perkuliahan.
5. Sahabat dan rekan saya Dwi Lastomo, Rian Wahyu, Suharto, Dwi Didit, Taufiq Ramadhan, Wandi, Bapak Aji, Bapak Andi Imran, Bapak Jamal, Bapak Izza, Bapak Yose, Bapak Darman, Bapak Fakhruddin, teman-teman e51, dan teman-teman laboratorium PSOC yang bersedia berdiskusi dan memberikan saran maupun kritik dalam Tugas Akhir ini.



6. Mas Herlambang Setiadi selaku penasihat saya yang telah memberikan pengarahan dan motivasi kepada penulis selama penulis menyelesaikan tugas akhir.
7. Sahabat dan rekan-rekan HIMATEKTRO ITS 2012/2013, BEM ITS 2014/2015, MM ITS 2015, serta seluruh jajaran pengurus KAMMI Sepuluh Nopember yang selalu memotivasi saya ketika saya sedang dalam proses pengerjaan Tugas Akhir.
8. Semua pihak yang telah membantu penulis dalam menyelesaikan Tugas Akhir yang tidak dapat penulis sebutkan.

Penulis berharap semoga apa yang ada dalam tugas akhir ini dapat memberikan manfaat kepada pembaca terutama mahasiswa Teknik Elektro.

Surabaya, Desember 2015

Penulis

# DAFTAR ISI

	Halaman
<b>JUDUL</b>	i
<b>LEMBAR PERNYATAAN</b>	iii
<b>LEMBAR PENGESAHAN</b>	v
<b>ABSTRAK</b>	vii
<b>ABSTRACT</b>	ix
<b>KATA PENGANTAR</b>	xi
<b>DAFTAR ISI</b>	xv
<b>DAFTAR GAMBAR</b>	xvii
<b>DAFTAR TABEL</b>	
<b>BAB 1 PENDAHULUAN</b>	1
1.1. Latar Belakang	1
1.2. Perumusan Masalah	3
1.3. Tujuan	3
1.4. Batasan Masalah	3
1.5. Metodologi	4
1.6. Sistematika Penulisan	5
1.7. Relevansi	5
<b>BAB 2 PEMODELAN SISTEM DAN ALGORITMA</b>	7
2.1. Pendahuluan	7
2.2. Pemodelan Motor DC	7
2.3. Prinsip Kerja Motor PMDC	10
2.3.1. Pemodelan Sebagai Motor	10
2.3.2. Pemodelan Sebagai Generator	10
2.4. Struktur dan Cara Kerja Motor PMDC	10
2.4.1. Struktur Rotor dan Stator	10
2.4.2. Gaya Gerak Listrik (GGL/emf) terinduksi dan Komutasi	11
2.4.3. Rangkaian Ekuivalen Motor PMDC	11
2.5. Prinsip Pengaturan Kecepatan Motor PMDC	13
2.5.1. Persamaan Fundamental	14
2.5.2. Pengaturan Jangkar	14
2.6. Kontroler PI	15
2.6.1. Kontroler Proporsional	15
2.6.2. Kontroler Integral	16



2.6.3. Kontroler Proportional Integral	17
2.7. Particle Swarm Optimization (PSO)	18
<b>BAB 3 PENERAPAN <i>PARTICLE SWARM OPTIMIZATION</i> PADA <i>OPTIMAL CONTROL PERMANENT MAGNET DIRECT CURRENT MOTOR</i></b>	21
3.1. Algoritma Pengerjaan Tugas Akhir	21
3.2. Sistem Motor DC Magnet Permanen dengan Kontroler PI	22
3.2.1 Data Sistem	23
3.3. Pemodelan Linear Motor PMDC	25
3.3.1. Persamaan Elektrik Motor DC	25
3.3.2. Persamaan Mekanik Motor DC	27
3.3.3. Sistem dengan Penambahan Kontroler PI	28
3.3.4. Pemodelan Gabungan Skema Penggerak Motor PMDC dengan Kontroler PI	33
3.4. Koordinasi Kontroler PI menggunakan PSO	35
<b>BAB 4 SIMULASI DAN ANALISIS</b>	39
4.1. Paramater PSO Pada Sistem Motor PMDC	39
4.2. Respon Sistem dengan PI dan PI yang dioptimisasi dengan PSO	40
4.2.1. Respon Sistem dengan Kecepatan Referensi 100 rad/sec	41
4.2.1.1. Respon Tegangan ( $V_T$ ) Sistem Motor PMDC	41
4.2.1.2. Respon Arus ( $I_A$ ) Sistem Motor PMDC	44
4.2.1.3. Respon Torsi Elektrik Motor ( $T_E$ ) Sistem Motor PMDC	48
4.2.1.4 Respon Kecepatan ( $\omega_M$ ) Sistem Motor PMDC	52
4.2.2. Respon Sistem dengan Kecepatan Referensi 150 rad/sec	56
4.2.2.1. Respon Tegangan ( $V_T$ ) Sistem Motor PMDC	56
4.2.2.2. Respon Arus ( $I_A$ ) Sistem Motor PMDC	60
4.2.2.3. Respon Torsi Elektrik Motor ( $T_E$ ) Sistem Motor PMDC	64
4.2.2.4 Respon Kecepatan ( $\omega_M$ ) Sistem Motor PMDC	68
<b>BAB 5 PENUTUP</b>	73
5.1. Kesimpulan	73
5.2. Saran	73

**DAFTAR PUSTAKA**

**INDEKS**

**DAFTAR SIMBOL DAN SINGKATAN**

**ADDENDUM**

**RIWAYAT HIDUP PENULIS**

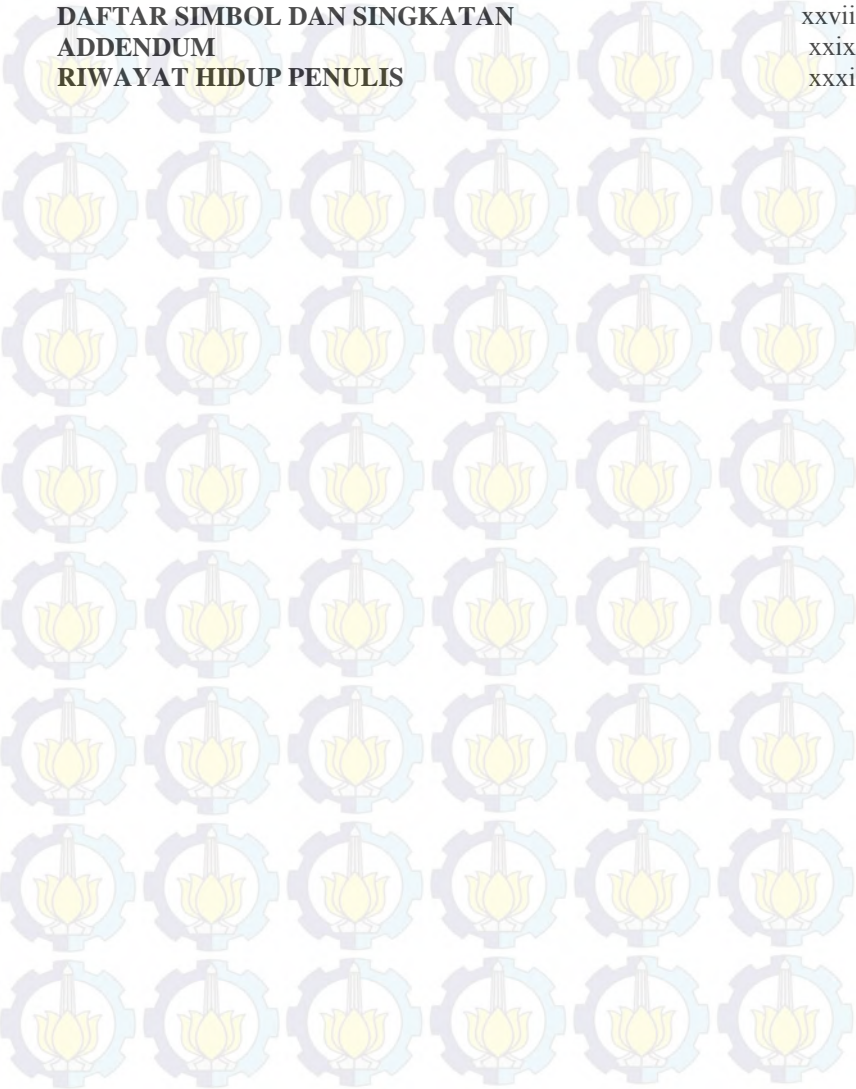
xxi

xxv

xxvii

xxix

xxxix



## DAFTAR TABEL

	Halaman
<b>Tabel 3.1</b> Parameter Sistem	24
<b>Tabel 3.2</b> Parameter Kontroler PI	33
<b>Tabel 3.3</b> Parameter PSO	36
<b>Tabel 3.4</b> Hasil optimisasi parameter PI menggunakan PSO	37
<b>Tabel 4.1</b> Parameter PSO	40
<b>Tabel 4.2</b> Perbandingan <i>overshoot</i> dan <i>settling time</i> respon tegangan motor PMDC pada 0-25 detik ( $\omega_{ref} = 100$ rad/sec)	42
<b>Tabel 4.3</b> Perbandingan <i>overshoot</i> dan <i>settling time</i> respon tegangan motor PMDC saat 25-50 detik ( $T_L = 1$ Nm, $\omega_{ref} = 100$ rad/sec)	43
<b>Tabel 4.4</b> Perbandingan <i>overshoot</i> dan <i>settling time</i> respon tegangan motor PMDC saat 25-50 detik ( $T_L = 2$ Nm, $\omega_{ref} = 100$ rad/sec)	44
<b>Tabel 4.5</b> Perbandingan <i>overshoot</i> dan <i>settling time</i> respon Arus Motor PMDC pada 0-25 detik ( $\omega_{ref} = 100$ rad/sec)	46
<b>Tabel 4.6</b> Perbandingan <i>overshoot</i> dan <i>settling time</i> respon Arus motor PMDC saat 25-50 detik ( $T_L = 1$ Nm, $\omega_{ref} = 100$ rad/sec)	46
<b>Tabel 4.7</b> Perbandingan <i>overshoot</i> dan <i>settling time</i> respon arus motor PMDC saat 25-50 detik ( $T_L = 2$ Nm, $\omega_{ref} = 100$ rad/sec)	48
<b>Tabel 4.8</b> Perbandingan <i>overshoot</i> dan <i>settling time</i> respon torsi elektrik motor PMDC saat 0-25 ( $\omega_{ref} = 100$ rad/sec)	50
<b>Tabel 4.9</b> Perbandingan <i>overshoot</i> dan <i>settling time</i> respon torsi elektrik motor PMDC saat 25-50 detik ( $T_L = 1$ Nm, $\omega_{ref} = 100$ rad/sec)	50
<b>Tabel 4.10</b> Perbandingan <i>overshoot</i> dan <i>settling time</i> respon torsi elektrik motor PMDC saat 25-50 detik ( $T_L = 2$ Nm, $\omega_{ref} = 100$ rad/sec)	52
<b>Tabel 4.11</b> Perbandingan <i>overshoot</i> dan <i>settling time</i> respon Kecepatan motor PMDC saat 0-25 ( $\omega_{ref} = 100$	53

	rad/sec)	
<b>Tabel 4.12</b>	Perbandingan <i>overshoot</i> dan <i>settling time</i> respon kecepatan motor PMDC saat 25-50 detik ( $T_L = 1$ Nm, $\omega_{ref} = 100$ rad/sec)	54
<b>Tabel 4.13</b>	Perbandingan <i>overshoot</i> dan <i>settling time</i> respon kecepatan motor PMDC saat 25-50 detik ( $T_L = 2$ Nm, $\omega_{ref} = 100$ rad/sec)	56
<b>Tabel 4.14</b>	Perbandingan <i>overshoot</i> dan <i>settling time</i> respon tegangan motor PMDC pada 0-25 detik ( $\omega_{ref} = 120$ rad/sec)	58
<b>Tabel 4.15</b>	Perbandingan <i>overshoot</i> dan <i>settling time</i> respon tegangan motor PMDC saat 25-50 detik ( $T_L = 1$ Nm, $\omega_{ref} = 120$ rad/sec)	59
<b>Tabel 4.16</b>	Perbandingan <i>overshoot</i> dan <i>settling time</i> respon tegangan motor PMDC saat 25-50 detik ( $T_L = 2$ Nm, $\omega_{ref} = 120$ rad/sec)	60
<b>Tabel 4.17</b>	Perbandingan <i>overshoot</i> dan <i>settling time</i> respon Arus Motor PMDC hingga pada 0-25 detik ( $\omega_{ref} = 120$ rad/sec)	62
<b>Tabel 4.18</b>	Perbandingan <i>overshoot</i> dan <i>settling time</i> respon arus motor PMDC saat 25-50 detik ( $T_L = 1$ Nm, $\omega_{ref} = 120$ rad/sec)	62
<b>Tabel 4.19</b>	Perbandingan <i>overshoot</i> dan <i>settling time</i> respon arus motor PMDC saat 25-50 detik ( $T_L = 2$ Nm, $\omega_{ref} = 120$ rad/sec)	64
<b>Tabel 4.20</b>	Perbandingan <i>overshoot</i> dan <i>settling time</i> respon torsi elektrik motor PMDC saat 0-25 detik ( $\omega_{ref} = 120$ rad/sec)	65
<b>Tabel 4.21</b>	Perbandingan <i>overshoot</i> dan <i>settling time</i> respon torsi elektrik motor PMDC saat 25-50 detik ( $T_L = 1$ Nm, $\omega_{ref} = 120$ rad/sec)	66
<b>Tabel 4.22</b>	Perbandingan <i>overshoot</i> dan <i>settling time</i> respon torsi elektrik motor PMDC saat 25-50 detik ( $T_L = 2$ Nm, $\omega_{ref} = 120$ rad/sec)	67
<b>Tabel 4.23</b>	Perbandingan <i>overshoot</i> dan <i>settling time</i> respon kecepatan motor PMDC saat 0-25 ( $\omega_{ref} = 120$	69



rad/sec)

<b>Tabel 4.24</b>	Perbandingan <i>overshoot</i> dan <i>settling time</i> respon kecepatan motor PMDC saat 25-50 detik ( $T_L = 1$ Nm, $\omega_{ref} = 120$ rad/sec)	70
-------------------	---	----

<b>Tabel 4.25</b>	Perbandingan <i>overshoot</i> dan <i>settling time</i> respon kecepatan motor PMDC saat 25-50 detik ( $T_L = 2$ Nm, $\omega_{ref} = 120$ rad/sec)	72
-------------------	---	----



## DAFTAR SIMBOL DAN SINGKATAN

PMDC	: <i>Permanent Magnet Direct Current</i>
PI	: <i>Proportional Integral</i>
PSO	: <i>Particle Swarm optimization</i>
GGL	: Gaya Gerak Listrik
$I_F$	: arus medan
$V_F$	: tegangan medan
$R_F$	: resistansi medan
$L_F$	: induktansi medan
$I_A$	: arus jangkar
$E_B$	: <i>back emf</i>
$V_T$	: tegangan terminal
$R_A$	: resistansi jangkar
$T_E$	: torsi elektrik / torsi motor
$T_L$	: torsi beban / torsi mekanik
$B_M$	: koefisien redaman sistem mekanis
$\omega_M$	: kecepatan sudut motor
$e$	: tegangan induksi
$\phi_F$	: fluks medan
$I_F$	: arus medan
$P_{dev}$	: <i>developed power</i>
$T_{dev}$	: torka yang dihasilkan pada mesin
$P_{best}$	: posisi terbaik yang dicapai partikel
$G_{best}$	: <i>Pbest</i> terbaik dalam populasi
$P$	: daya mekanik motor
$K_B$	: konstanta <i>back emf</i>
$K_T$	: konstanta torsi
$N$	: kecepatan dalam RPM (putaran per menit)
$E$	: gaya elektromagnetik
$K_p$	: konstanta <i>proportional</i>
$K_i$	: konstanta <i>integral</i>
$\omega_{ref}$	: kecepatan referensi

## DAFTAR GAMBAR

	Halaman
<b>Gambar 2.1</b> Rangkaian ekivalen motor DC Penguatan Terpisah ( <i>Separately Excited</i> )	8
<b>Gambar 2.2</b> Rangkaian ekivalen motor PMDC	12
<b>Gambar 2.3</b> Diagram blok kontroler proporsional	15
<b>Gambar 2.4</b> Diagram blok kontroler integral	16
<b>Gambar 2.5</b> Diagram blok kontroler PI	17
<b>Gambar 2.6</b> Flowchart koordinasi PSO	20
<b>Gambar 3.1</b> Flowchart pengerjaan tugas akhir	21
<b>Gambar 3.2</b> Skema motor PMDC dengan kontroler PI	23
<b>Gambar 3.3</b> Rangkaian ekivalen motor PMDC	25
<b>Gambar 3.4</b> Pemodelan Motor PMDC Melalui Persamaan Elektrik	26
<b>Gambar 3.5</b> Pemodelan Motor PMDC Melalui Persamaan Mekanik	28
<b>Gambar 3.6</b> Diagram Blok Plant Orde Kedua Tanpa Delay	30
<b>Gambar 3.7</b> Diagram Blok Kontroler PI	30
<b>Gambar 3.8</b> Diagram Blok Kontroler PI yang Diterapkan Pada Plant Orde Kedua Tanpa Delay	30
<b>Gambar 3.9</b> Diagram Blok Sistem Pengaturan Motor PMDC	32
<b>Gambar 3.10</b> Diagram Skematik Motor PMDC	35
<b>Gambar 4.1</b> Respon Tegangan Motor PMDC dengan Gangguan Torsi Mekanik 1 Nm ( $\omega_{ref} = 100$ rad/sec)	41
<b>Gambar 4.2</b> Respon Tegangan Motor PMDC dengan Gangguan Torsi Mekanik 2 Nm ( $\omega_{ref} = 100$ rad/sec)	43
<b>Gambar 4.3</b> Respon Arus Motor PMDC dengan Gangguan Torsi Mekanik 1 Nm ( $\omega_{ref} = 100$ rad/sec)	45
<b>Gambar 4.4</b> Respon Arus Motor PMDC dengan Gangguan Torsi Mekanik 2 Nm ( $\omega_{ref} = 100$ rad/sec)	47
<b>Gambar 4.5</b> Respon Torsi Elektrik Motor PMDC dengan Gangguan Torsi Mekanik 1Nm ( $\omega_{ref} = 100$ rad/sec)	49
<b>Gambar 4.6</b> Respon Torsi Elektrik Motor PMDC dengan Gangguan Torsi Mekanik 2 Nm ( $\omega_{ref} = 100$ rad/sec)	51
<b>Gambar 4.7</b> Respon Kecepatan Motor PMDC dengan Gangguan	

	Torsi Mekanik 1 Nm ( $\omega_{ref} = 100$ rad/sec)	53
<b>Gambar 4.8</b>	Respon Kecepatan Motor PMDC dengan Gangguan	
	Torsi Mekanik 2 Nm ( $\omega_{ref} = 100$ rad/sec)	55
<b>Gambar 4.9</b>	Respon Tegangan Motor PMDC dengan Gangguan	
	Torsi Mekanik 1 Nm ( $\omega_{ref} = 120$ rad/sec)	57
<b>Gambar 4.10</b>	Respon Tegangan Motor PMDC dengan Gangguan	
	Torsi Mekanik 2 Nm ( $\omega_{ref} = 120$ rad/sec)	59
<b>Gambar 4.11</b>	Respon Arus Motor PMDC dengan Gangguan Torsi	
	Mekanik 1 Nm ( $\omega_{ref} = 120$ rad/sec)	61
<b>Gambar 4.12</b>	Respon Arus Motor PMDC dengan Gangguan Torsi	
	Mekanik 2 Nm ( $\omega_{ref} = 120$ rad/sec)	63
<b>Gambar 4.13</b>	Respon Torsi Elektrik Motor PMDC dengan	
	Gangguan Torsi Mekanik 1 Nm ( $\omega_{ref} = 120$ rad/sec)	65
<b>Gambar 4.14</b>	Respon Torsi Elektrik Motor PMDC dengan	
	Gangguan Torsi Mekanik 2 Nm ( $\omega_{ref} = 120$ rad/sec)	67
<b>Gambar 4.15</b>	Respon Kecepatan Motor PMDC dengan Gangguan	
	Torsi Mekanik 1 Nm ( $\omega_{ref} = 120$ rad/sec)	68
<b>Gambar 4.16</b>	Respon Kecepatan Motor PMDC dengan Gangguan	
	Torsi Mekanik 2 Nm ( $\omega_{ref} = 120$ rad/sec)	71



# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Pertumbuhan sektor industri saat ini pada industri yang berskala besar, menengah, maupun industri-industri yang berskala kecil cukup pesat. Perkembangan dunia mesin listrik pada saat ini juga cukup pesat pada dunia industri. Salah satu mesin listrik yang pada saat ini berkembang pesat di dunia industri adalah motor arus searah (motor DC). Motor DC telah ada selama lebih dari seabad. Keberadaan motor DC telah membawa perubahan besar sejak dikenalkan motor induksi, atau sering disebut *AC Shunt Motor*. Mesin listrik dapat berfungsi sebagai generator listrik apabila di dalam mesin listrik tersebut terjadi proses konversi energi mekanik menjadi energi listrik, dan juga dapat berfungsi sebagai motor listrik apabila didalam mesin listrik tersebut terjadi proses konversi dari energi listrik menjadi energi mekanik. Motor listrik merupakan perangkat elektromagnetik yang mengubah energi listrik menjadi energi mekanik. Energi mekanik ini digunakan untuk memutar *impeller, pump, fan*, atau blower, menggerakkan kompresor, dan mengangkat bahan. Motor listrik juga digunakan di peralatan rumah tangga (mixer, bor listrik, kipas angin) dan di dunia industri. Motor listrik adalah “kuda kerja” nya industri sebab motor di dunia industri diperkirakan menggunakan sekitar 70% beban listrik total di industri. [3]

Motor DC adalah jenis motor yang banyak digunakan di dunia elektronik, industri, dan komponen pendukung lainnya untuk beberapa peralatan atau instrumentasi elektronik. Aplikasi industri motor DC yang digunakan dalam industri pada umumnya memiliki kapasitas daya yang relatif dan disesuaikan dengan volume produksi dan beban mekanis. Motor DC termasuk dalam kategori jenis motor yang paling banyak digunakan baik dalam lingkungan industri, peralatan rumah tangga, dan mainan anak-anak maupun sebagai piranti pendukung sistem instrumenstasi elektronik. [4]

Motor DC memerlukan suplai tegangan searah pada kumparan jangkar dan kumparan medan untuk mengubah energi listrik menjadi energi mekanik. Kumparan medan pada motor DC disebut stator (bagian yang tidak berputar) dan kumparan jangkar disebut rotor (bagian yang berputar). Motor DC sering dimanfaatkan sebagai penggerak pintu geser otomatis dan dalam aplikasi rangkaian robot sederhana. [5]

Dalam dunia industri, sistem produksi memerlukan kecepatan dan ketepatan namun konsumsi daya yang efisien sangat diperlukan. Motor DC sering digunakan dalam memproduksi, mengangkat, dan menurunkan ataupun menggerakkan perangkat mesin namun tidak setiap saat motor DC bekerja dengan beban kerja yang tetap. Hal tersebut akan mempengaruhi kecepatan putaran awal dari motor tersebut jika motor diberi gangguan beban. Semakin berat beban yang diberikan, maka kecepatan putaran motor akan berkurang dari kecepatan sebelumnya dan begitu juga sebaliknya, jika beban yang diberikan berkurang maka kecepatan putaran motor akan bertambah sehingga diperlukan operator untuk menambah atau mengurangi tegangan suplai ke motor DC sesuai dengan beban yang dikerjakan. [11]

Seiring dengan kemajuan teknologi, permasalahan pada dunia industri tentang ketidakstabilan dari kecepatan motor DC ketika awal penyalan sangatlah kompleks sehingga ketika motor DC tersebut bekerja dalam suatu proses yang membutuhkan kecepatan yang konstan, maka sistem tersebut akan terganggu. Oleh karena itu penggunaan kontrol dalam dunia industri sangatlah penting, terutama pada pengaturan kecepatan motor. Pengaturan kecepatan motor dapat dilakukan dengan menambah atau mengurangi tegangan suplai ke motor DC tersebut, namun akan mengakibatkan pekerjaan yang lebih rumit dan waktu yang lebih lama, karena harus menyeleksi waktu untuk mempercepat atau memperlambat kecepatan motor. [2]

Motor DC memiliki banyak manfaat dalam kehidupan sehari-hari dan dalam dunia industri serta memudahkan proses pekerjaan sehingga proses industri dapat berjalan baik. Oleh karena itu sangat penting untuk mengetahui dan mengerti pengertian motor DC, jenis-jenis motor DC, prinsip kerja, aplikasi, dan perhitungan motor DC. [8]

Peralatan-peralatan industri seperti motor DC sangat membutuhkan suplai tegangan yang sesuai agar kecepatan yang diperoleh sesuai dengan yang dibutuhkan. Dengan menggunakan proses tala pada kontroler PI yang dioptimisasi dengan metode PSO nantinya diharapkan diperoleh performansi sistem yang diinginkan. [2]

## **1.2. Perumusan Masalah**

Permasalahan yang dibahas pada tugas akhir ini adalah tentang pengaturan dengan kontroler PI dan juga optimisasi kontrol motor PMDC menggunakan algoritma PSO untuk mendapatkan performansi motor PMDC yang sesuai agar kecepatan yang diperoleh sesuai dengan yang dibutuhkan.



### 1.3. Tujuan

Tujuan dari tugas akhir ini adalah mendapatkan performansi sistem yang optimal. Sistem motor PMDC dikontrol oleh kontroler PI sehingga error pada sistem dapat diminimalisasi sehingga akan diperoleh performansi sistem yang diharapkan. Melalui proses setting dan penalaan kontroler PI yang dioptimisasi dengan algoritma PSO untuk pengaturan motor PMDC diharapkan akan diperoleh hasil yang paling optimal dan akhirnya dapat digunakan sebagai salah satu acuan atau masukan untuk penyelesaian kasus yang sama.

Hasil yang diperoleh dari tugas akhir ini diharapkan dapat memberikan manfaat dan referensi bagi dunia penelitian dalam meneliti kasus motor PMDC dan untuk mendapatkan performansi sistem motor PMDC yang diinginkan. Selain itu juga dapat menjadi referensi bagi mahasiswa lain yang hendak mengembangkan solusi dari kendala-kendala yang ada untuk tugas akhirnya.

### 1.4 Batasan Masalah

Batas masalah dalam Tugas Akhir ini adalah :

1. Model sistem motor PMDC yang digunakan dalam Tugas Akhir ini diperoleh dari paper “*State Space Modeling and Eigenvalue Analysis of the Permanent Magnet DC Motor Drive System*” IEEE 2010.
2. Tugas Akhir hanya sebatas simulasi. Simulasi dilakukan dengan menggunakan software Matlab dan Simulink.
3. Parameter Motor PMDC diperoleh dari paper “*Sliding Mode Speed Control Of A DC Motor*”, *International Conference on Communication Systems and Network Technologies*, IEEE 2011.
4. Kontroler yang digunakan adalah PI.
5. *Set Point* kecepatan motor PMDC adalah 100 rad/sec dan dibatasi hingga 120 rad/sec.
6. Konverter dianggap ideal sehingga dimodelkan dengan konstanta *gain* 1.
7. Motor diberi beban saat mencapai waktu 25 detik.
8. Tegangan motor ( $V_T$ ) dibatasi hingga 103,35 V.
9. Faktor ekonomi tidak diperhitungkan.
10. Faktor ripple diabaikan.
11. Rugi daya diabaikan.

## 1.5. Metodologi

Dalam menyelesaikan Tugas Akhir ini, metode yang digunakan dalam menjawab permasalahan yang akan diselesaikan adalah sebagai berikut :

### 1. Studi Literatur

Pada tahap studi literatur, dilakukan pencarian data-data dari paper, jurnal ilmiah, artikel, dan berbagai sumber lainnya. Studi yang dilakukan dengan mempelajari tentang PMDC, Motor DC, dan proses penalaan pada kontroler PI. Selain itu juga dipelajari algoritma *Particle Swarm Optimization* (PSO) untuk mengatur optimisasi kontrol motor PMDC dan pengimplementasian algoritma tersebut.

### 2. Pemodelan Sistem

Pada tahap ini dilakukan pengolahan data ke dalam suatu persamaan matematis dan pemodelan *PMDC motor drive system* menggunakan simulink.

### 3. Simulasi

Pada tahap ini dilakukan simulasi dengan pemodelan sistem yang telah dirancang dengan menggunakan software matlab dan simulink. Proses penalaan pada kontroler PI dengan menggunakan algoritma PSO nantinya diharapkan diperoleh performansi sistem yang diinginkan.

### 4. Analisis Data

Pada tahap ini dilakukan analisis data hasil uji coba yang telah dikumpulkan untuk mendapatkan efisiensi serta performansi sistem yang optimal. Dalam proses optimisasi kontrol motor PMDC, hal-hal yang perlu diperhatikan adalah tingkat kompleksitas sistem perindustrian yang semakin tinggi karena kemajuan teknologi, kondisi peralatan industri seperti motor DC, serta suplai tegangan yang sesuai agar kecepatan yang diperoleh sesuai dengan yang dibutuhkan [5]. Dengan suplai tegangan yang sesuai, maka performansi sistem akan tetap terjaga secara optimal sehingga dapat diperoleh performansi sistem yang diinginkan.

### 5. Kesimpulan

Pada tahap kesimpulan, diperoleh hasil dari penelitian yang telah dilakukan.

### 6. Penulisan Buku Tugas Akhir

Sebagai tahap akhir dari proses penelitian, maka penulis membuat laporan tertulis berupa hasil analisis dari permasalahan dan disusun dalam bentuk buku Tugas Akhir.

### 1.6. Sistematika Penulisan

Sistematika penulisan Tugas Akhir ini terbagi atas lima bagian yaitu sebagai berikut :

1. BAB 1 merupakan pendahuluan yang membahas mengenai latar belakang, rumusan masalah, tujuan, batasan masalah, metode penelitian, sistematika penulisan, serta relevansi dari Tugas Akhir.
2. BAB 2 merupakan dasar teori yang merupakan penjelasan teori tentang motor PMDC, kontroler PI, dan juga algoritma *Particle Swarm Optimization* (PSO).
3. BAB 3 merupakan penjelasan teori tentang optimisasi kontrol motor PMDC menggunakan algoritma PSO serta permodelan sistem motor PMDC agar dapat disimulasikan menggunakan program komputer MATLAB-SIMULINK.
4. BAB 4 merupakan hasil simulasi dari percobaan yang telah dilakukan.
5. BAB 5 merupakan kesimpulan dari Tugas Akhir yang mengemukakan kesimpulan dari serangkaian proses pembahasan dan mengemukakan saran-saran yang berhubungan dengan pokok-pokok pembahasan.

### 1.7. Relevansi

Hasil yang diperoleh dari Tugas Akhir ini diharapkan dapat memberikan manfaat terhadap perkembangan sistem kelistrikan, khususnya bagi sistem kelistrikan yang terpasang pada suatu daerah agar nantinya diperoleh performansi sistem yang optimal. Hasil yang diperoleh juga berguna untuk meningkatkan penguasaan Ilmu Pengetahuan dan Teknologi (IPTEK) di bidang sistem tenaga bagi pengusul Tugas Akhir dan berbagai macam penelitian lainnya. Selain itu hasil dari penelitian ini juga dapat menjadi referensi bagi mahasiswa lain yang hendak mengambil masalah yang serupa sebagai Tugas Akhir atau penelitian.



## BAB 2

### PEMODELAN SISTEM DAN ALGORITMA

#### 2.1 Pendahuluan

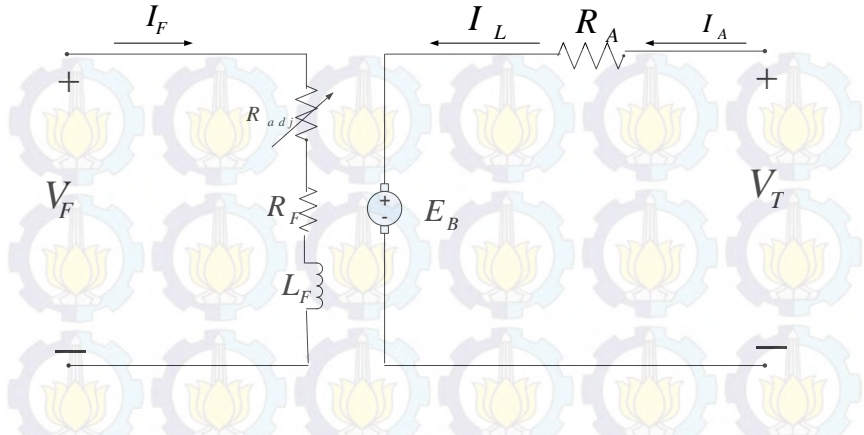
Motor arus searah (dc) adalah jenis motor yang pertama kali digunakan dalam dunia industri karena karakteristik pengaturannya yang baik. Kumparan medan pada motor dc terletak di stator sedangkan kumparan jangkar terletak di rotor. Pada prinsip kerja motor dc, arah medan magnet rotor akan selalu berlawanan dengan arah medan magnet stator, hal tersebut karena disebabkan oleh sifat magnet yaitu bahwa jika magnet yang berlawanan arah didekatkan satu sama lain mereka akan saling tarik menarik dan sebaliknya jika magnet yang searah akan saling tolak. Arah medan magnet stator adalah tetap sehingga arah medan magnet rotor harus menyesuaikan atau dirubah dengan tujuan untuk menjaga kontinuitas momen putar rotor. Agar tercipta efek perubahan arah medan rotor, arah aliran arus yang mengalir dalam rangkaian jangkar diubah arahnya dengan cara menghubungkan sumber tegangan luar dengan rangkaian jangkar melalui sikat (brush) yang dilengkapi dengan komutator. Fungsi cincin komutasi tersebut adalah untuk menjaga agar posisi medan jangkar selalu optimal dalam menghasilkan medan putar. Sedangkan medan stator dapat dibangkitkan dengan magnet permanen atau elektromagnetis [3].

#### 2.2 Pemodelan Motor DC

Persamaan matematika pada pemodelan mesin dc dapat diturunkan dengan menggunakan hukum-hukum dasar rangkaian listrik serta prinsip-prinsip hubungan elektromekanik melalui rangkaian ekivalen motor dc. Persamaan tersebut digunakan untuk analisis dan perencanaan teknik pengaturan motor [3]. Rangkaian ekivalen sebuah mesin dc dengan penguatan terpisah (*separately excited*) didefinisikan seperti pada Gambar 2.1.

$I_F$  arus medan  
 $V_F$  tegangan medan  
 $R_F$  resistansi medan  
 $L_F$  induktansi medan

$I_A$  arus jangkar  
 $E_B$  *back-emf*  
 $V_T$  tegangan terminal  
 $R_A$  resistansi jangkar



**Gambar 2.1** Rangkaian ekivalen motor DC Penguatan Terpisah  
(*Separately Excited*)

Persamaan yang mengatur kerja motor DC dapat dituliskan sebagai berikut. Persamaan rangkaian medan ditunjukkan pada persamaan (2.1) sedangkan persamaan rangkaian jangkar pada persamaan (2.2). Fungsi terhadap waktu ditunjukkan melalui besaran yang ditulis dengan huruf kecil.

$$V_F = R_F I_F + L_F \frac{dI_F}{dt} \quad (2.1)$$

$$V_T = R_A I_A + E_B \quad (2.2)$$

$$E_B = K I_F \omega \quad (2.3)$$

$$T_E = K I_F I_A \quad (2.4)$$

Hubungan elektromekanik dituliskan sebagai berikut:

$$T_E = T_L + J_M \frac{d\omega_M}{dt} + B_M \omega_M \quad (2.5)$$



$T_E$  dan  $T_L$  merupakan torsi motor dan torsi beban, sedangkan  $J_M$  adalah konstanta inersia dan  $\omega_M$  merupakan kecepatan motor serta  $B_M$  adalah koefisien redaman sistem mekanis.

Pemodelan dari rangkaian medan, rangkaian jangkar, model elektromekanis, *back emf* dari mesin dc ditunjukkan pada persamaan (2.1), (2.2), (2.3), (2.4), (2.5).

Rangkaian pada stator direpresentasikan oleh hambatan  $R_F$  dan induktor  $L_F$  yang tersusun secara seri. Pada rangkaian DC, induktor berlaku sebagai short circuit sehingga besar tegangan pada stator tersebut adalah :

$$V_F = I_F R_F \quad (2.6)$$

Motor DC terdiri dari berbagai bagian dan komponen utama, diantaranya sebagai berikut:

- Kutub medan. Motor DC memiliki 2 kutub yaitu kutub utara dan kutub selatan. Pada kutub medan ini garis magnetik energi membesar melintasi ruang terbuka diantara kutub-kutub dari kutub utara ke selatan. Untuk motor dc yang lebih kompleks terdapat satu atau lebih elektromagnet.
- Dinamo, bentuknya silinder, berfungsi untuk menggerakkan beban. Pada motor dc yang sederhana, dinamo berputar pada medan magnet yang dibentuk oleh kutub-kutub.
- Commutator, berfungsi untuk mentransmisikan arus antara sumber daya dan dinamo [17].

Motor dc juga berfungsi sebagai pengendali kecepatan, tanpa mempengaruhi pasokan kualitas daya. Dengan meningkatkan tegangan dinamo maka akan meningkatkan kecepatan motor dc. Selain itu, untuk meningkatkan kecepatan juga dapat dilakukan dengan menurunkan arus medan [22].

Hubungan antara kecepatan, flux medan, dan tegangan dinamo ditunjukkan dalam persamaan berikut ini :

Gaya Elektomagnetik (E) :

$$E = K_B \phi N \quad (2.7)$$

$$T = K_T \phi I_a \quad (2.8)$$

#### Keterangan

$E$  : gaya elektromagnetik yang dikembangkan pada terminal dinamo (volt)

$\Phi$  : flux medan yang berbanding lurus dengan arus medan

$N$  : kecepatan dalam RPM (putaran per menit)

$T$  : torsi elektromagnetik

$I_a$  : arus dinamo

$K_B$  : konstanta persamaan *back-emf*

$K_T$  : konstanta persamaan torsi

### 2.3 PRINSIP KERJA MOTOR PMDC

#### 2.3.1 Pemodelan sebagai Motor

Jika terdapat beban gaya pada batang (*load*) yang berlawanan terhadap arah gaya Lorentz, maka beban tersebut akan memperlambat batang sehingga menimbulkan keseimbangan gaya antara gaya beban terhadap gaya Lorentz. Saat kondisi ini tercapai, batang akan bergerak dengan kecepatan konstan [11].

Besar daya yang digunakan sebagai input pada batang berasal dari sumber tegangan  $V_T$  yaitu :

$$P = V_T \cdot I_A \quad (2.9)$$

$P$  : daya input

$V_T$  : tegangan sumber

$I_A$  : arus jangkar

#### 2.3.2 Pemodelan sebagai Generator

Jika terdapat gaya yang menarik batang pada saat besar arus dalam rangkaian bernilai nol, maka batang akan menghasilkan beda tegangan induksi pada kedua ujungnya. Beda tegangan ini menghasilkan arus yang berlawanan arah jarum jam. Arus ini dapat mengisi baterai atau hilang sebagai panas pada resistor. Dalam kondisi ini, telah terjadi perubahan energi mekanis (akibat tarikan pada batang) menjadi energi kimiawi (muatan yang tersimpan pada senyawa kimia dalam baterai) [5].

### 2.4 STRUKTUR DAN CARA KERJA MOTOR PMDC

#### 2.4.1 Struktur Rotor dan Stator

Secara umum, motor DC terdiri dari bagian stator (bagian diam) berbentuk silindris dengan magnet yang dipasang secara berpasangan (*poles*, magnet pada motor PMDC berupa magnet permanen), dan terdapat beda kutub magnet pada keliling stator [6].

Didalam stator terdapat rotor (bagian yang bergerak) yang terdiri dari silinder besi terlamina yang dipasang pada batang penggerak (*shaft*) yang disokong oleh bantalan (*bearing*) sehingga membuat shaft dapat berputar. Ruang kosong antara permukaan rotor dengan stator dapat diberikan lilitan *armature* untuk membesarkan efek gaya Lorentz [7].

Fluks magnet akan cenderung memilih jalur yang paling kecil reluktansinya. Hal itu disebabkan karena reluktansi udara lebih besar dibanding reluktansi pada besi, maka fluks akan bergerak melewati lintasan terpendek dari stator ke rotor. Dan pada akhirnya fluks medan magnet akan berarah tegak lurus secara radial terhadap kumparan *armature*. Fluks magnet bernilai sama besar untuk sekeliling rotor karena besar medan magnet sama. Dengan hukum Lorentz dan aturan tangan kanan maka dihasilkan torka yang memutar shaft yang arahnya berlawanan arah jarum jam untuk kondisi yang digambarkan [8].

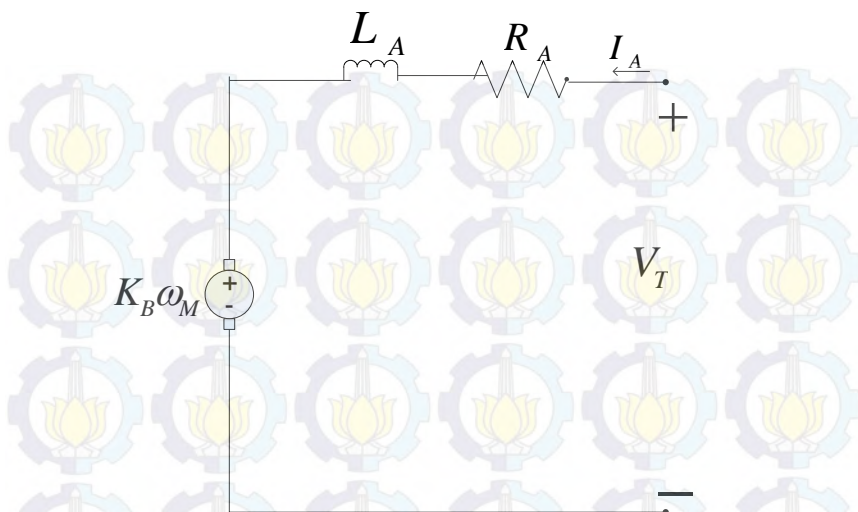
#### **2.4.2 Gaya Gerak Listrik (GGL/emf) terinduksi dan Komutasi**

Saat rotor berputar, konduktor (*armature* dan besi) bergerak melintasi medan magnet yang dihasilkan stator sehingga akan menimbulkan gaya Lorentz. Namun seiring perputaran, arah medan magnet yang dilintasi akan berubah terbalik. Hal tersebut menyebabkan timbulnya gaya yang melawan arah putaran. Hal ini dapat diatasi dengan menggunakan *komutator* untuk membalik arus sehingga arah medan magnet tetap sepanjang putaran rotor [11].

#### **2.4.3 Rangkaian Ekuivalen Motor PMDC**

Rangkaian ekuivalen motor DC magnet permanen (PMDC) sama dengan rangkaian motor DC biasa. Perbedaannya adalah pada motor DC magnet permanen tidak memiliki kumparan eksitasi atau kumparan medan. Kumparan medan pada motor PMDC berupa magnet permanen sehingga medan magnet yang dihasilkan berupa fluks magnetik konstan. Fluks magnetik yang konstan menyebabkan arus medan yang dihasilkan juga konstan. Rangkaian ekuivalen motor PMDC dapat ditunjukkan pada gambar 2.2 dibawah ini :





**Gambar 2.2** Rangkaian ekivalen motor PMDC

Motor DC yang digunakan pada Tugas Akhir ini adalah motor DC dengan magnet permanen. Pada motor DC magnet permanen terdapat dua buah magnet permanen sehingga timbul medan magnet diantara dua magnet tersebut. Di dalam medan magnet terdapat perputaran jangkar / rotor. Jangkar pada medan magnet tersebut terletak di tengah motor. Jangkar memiliki jumlah kutub yang ganjil dan terdapat lilitan pada setiap kutubnya. Lilitan tersebut terhubung ke bagian yang disebut komutator. Sikat (*brush*) yang terhubung ke kutub positif dan negatif memberikan daya ke lilitan dan menyebabkan kutub yang satu akan ditolak oleh magnet permanen yang berada di dekatnya dan lilitan lain akan ditarik ke magnet permanen yang lain. Hal ini menyebabkan jangkar berputar dan komutator mengubah lilitan yang mendapat pengaruh polaritas medan magnet sehingga selama kutub positif dan negatif motor diberi daya maka jangkar akan terus mengalami perputaran [3]. Kecepatan putar motor DC ( $N$ ) didefinisikan dengan persamaan :

$$N = \frac{V_T - I_A R_A}{K_B \phi} \quad (2.10)$$

### Keterangan

N	: Kecepatan Putar Motor DC
$K_B$	: Konstanta <i>back-emf</i>
$\phi$	: Fluks Magnet
$V_T$	: Tegangan Terminal
$I_A$	: Arus jangkar motor
$R_A$	: Hambatan Jangkar Motor

Tegangan  $E_B$  pada rotor mewakili tegangan rata-rata yang terinduksi pada *armature* karena disebabkan gerak dari konduktor relatif terhadap medan magnet. Tegangan  $E_B$  biasanya disebut ***back emf*** karena arahnya berlawanan dengan arah arus sumber dari luar motor. Resistor  $R_A$  merupakan hambatan dari lilitan *armature* serta hambatan dari sikat (pada komutator). Besar tegangan terinduksi pada *armature* adalah :

$$E_B = K_B \phi \omega_M \quad (2.11)$$

$K_B$  adalah konstanta *back-emf* yang bergantung pada desain motor,  $\phi$  merupakan fluks magnet yang dihasilkan oleh tiap kutub stator dan  $\omega_M$  merupakan kecepatan sudut rotor.

Dari persamaan gaya Lorentz maka diperoleh torka yang dihasilkan pada mesin sebesar :

$$T_{dev} = K_T \phi I_A \quad (2.12)$$

$I_A$  adalah arus pada lilitan *armature* sehingga diperoleh besar *developed power* yaitu besar daya yang diubah menjadi *mechanical power* :

$$P_{dev} = \omega_m T_{dev} \quad (2.13)$$

Besar daya ini adalah sama dengan daya listrik yang dihasilkan :

$$P_{dev} = V_T I_A \quad (2.14)$$

## 2.5 Prinsip Pengaturan Kecepatan Motor PMDC

Prinsip pengaturan kecepatan motor PMDC tergantung pada persamaan dasar emf. Torsi, flux, arus, emf yang terinduksi, dan kecepatan dinormalisasi untuk menampilkan karakteristik motor. Tipe pengaturan motor PMDC adalah pengaturan jangkar. Metode ini



digunakan untuk menghasilkan berbagai pengaturan kecepatan. Karakteristik kecepatan torsi motor berhubungan dengan kedua arah rotasi dan torsi dalam arah rotasi. Seperti proses operasi pada umumnya yang dikenal sebagai operasi empat kuadran, prinsip pengaturan kecepatan motor PMDC berhubungan dengan tegangan dan arus input untuk jangkar motor PMDC. Hal tersebut digunakan untuk menentukan besarnya daya yang diberikan [7].

### 2.5.1 Persamaan Fundamental

Persamaan tegangan induksi pada flux medan dan kecepatan diberikan sebagai berikut :

$$e = K\phi_f\omega_m \quad (2.15)$$

Dari substitusi diperoleh persamaan sebagai berikut :

$$\omega_m = \frac{e}{\phi_f} = \frac{V - I_a R_a}{\phi_f} \quad (2.16)$$

Masing-masing V dan  $I_a$  adalah tegangan dan arus jangkar.

### 2.5.2 Pengaturan Jangkar

Besarnya kecepatan dapat diturunkan dari persamaan :

$$\omega_m = (V - I_a R_a) \quad (2.17)$$

Oleh karena itu, diperlukan pengaturan jangkar untuk memvariasikan nilai kecepatan pada kondisi berbagai perubahan tegangan yang terjadi. Membalikkan tegangan yang diterapkan menyebabkan perubahan arah dari putaran motor. Keuntungan yang diperoleh dari proses pengaturan jangkar adalah arus jangkar dapat dikendalikan dengan cepat. Respon ditentukan oleh waktu jangkar yang konstan dan memiliki nilai yang sangat rendah. Sebaliknya, konstanta waktu medan setidaknya bernilai 10 sampai 100 kali lebih besar dari konstanta waktu jangkar. Besarnya konstanta waktu pada medan menyebabkan respon medan yang dikendalikan oleh penggerak motor menjadi lambat[2].

Pengaturan jangkar terbatas pada kecepatan dengan besaran yang terbatas pada banyaknya suplai tegangan dc dan isolasi belitan jangkar. Jika besarnya suplai tegangan DC divariasikan dari nol ke nilai nominalnya, maka kecepatan dapat dikontrol dari nol ke nilai nominalnya. Oleh karena itu, pengaturan jangkar sangat ideal diterapkan

pada kecepatan yang lebih rendah dari nilai kecepatan nominalnya, sedangkan pengaturan medan cocok diterapkan pada kecepatan yang lebih besar dari nilai kecepatan nominalnya [3].

## 2.6 Kontroler PI

Error dalam suatu sistem dapat menyebabkan sistem tersebut mengalami gangguan dan membuat sistem tidak stabil. Agar sistem tersebut stabil kembali, maka diperlukan suatu kontroler. Fungsi kontroler tersebut adalah untuk mereduksi sinyal error pada sistem. Suatu sistem kontrol akan semakin baik kinerjanya jika semakin cepat reaksi sistem mengikuti sinyal aktual dan semakin kecil kesalahan yang terjadi. Kontroler yang banyak digunakan saat ini adalah kontroler Proporsional Integral (PI). Penggunaan kontroler PI sederhana dan dapat meningkatkan performansi sistem dengan cepat. [10]

### 2.6.1 Kontroler Proporsional

Keluaran kontroler proporsional adalah hasil kali antara sinyal input dengan gain proporsional. Selain itu, keluarannya juga akan sebanding dengan besar sinyal error, yaitu selisih antara harga aktualnya dengan besaran yang *reference*. Perubahan sinyal input tersebut menyebabkan sistem mengubah keluaran sebesar konstanta pengali. [26]

Blok diagram hubungan antara besaran aktual, besaran *setting*, dan besaran keluaran kontroler proporsional ditunjukkan pada gambar 2.3. Sinyal positif atau negatif yang dikeluarkan kontroler dipengaruhi oleh sinyal *error*, yaitu selisih antara besaran *setting* dengan besaran aktual. Sinyal positif akan mempercepat pencapaian harga *setting*, sedangkan sinyal negatif akan memperlambat tercapainya harga atau performansi sistem yang diinginkan. [26]



**Gambar 2.3** Diagram blok kontroler proporsional

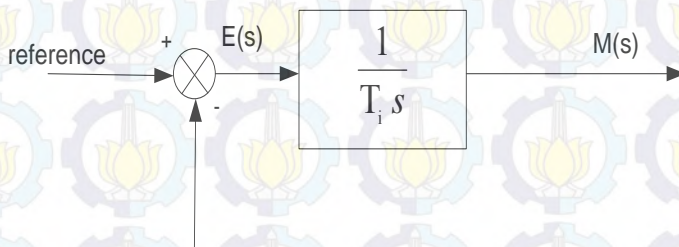
Konstanta proporsional pada kontroler proporsional adalah nilai faktor penguatan terhadap sinyal kesalahan  $K_p$ . Sistem respon yang

lambat akan terjadi apabila nilai  $K_p$  kecil karena kontroler proporsional hanya melakukan koreksi kesalahan yang kecil. Sebaliknya jika nilai  $K_p$  dinaikkan maka respon sistem akan menjadi lebih besar. Sedangkan jika nilai  $K_p$  diperbesar sehingga mencapai nilai yang berlebihan, maka akan terjadi osilasi pada respon sistem atau sistem tidak bekerja secara stabil. [10]

### 2.6.2. Kontroler Integral

Kontroler integral berfungsi untuk menghasilkan respon sistem yang memiliki kesalahan keadaan mantap nol. Jika sebuah *plant* memiliki unsur integrator ( $1/s$ ) maka akan terjadi keluaran sistem yang memiliki kesalahan keadaan mantap nol. Respon sistem yang demikian dapat diperbaiki dengan kontroler integral. [10]

Keluaran pada kontroler integral dipengaruhi oleh perubahan yang sebanding dengan sinyal kesalahan atau *error*, selain itu keluarannya adalah jumlah yang kontinyu dari perubahan inputnya. Keluaran akan tetap menjaga sistem dalam keadaan stabil sebelum terjadi perubahan masukan jika sinyal *error* tidak mengalami perubahan. Gambar 2.4 adalah blok diagram besaran kesalahan dengan keluaran kontroler integral.



**Gambar 2.4** Diagram blok kontroler integral

Jika terjadi kesalahan berlipat ganda maka nilai laju perubahan kontroler integral menjadi dua kali semula. Sinyal kesalahan yang relatif kecil dapat menyebabkan laju keluaran yang lebih besar apabila nilai konstanta integrator berubah menjadi lebih besar. [26]

Kontroler integral memiliki beberapa karakteristik yaitu sebagai berikut :



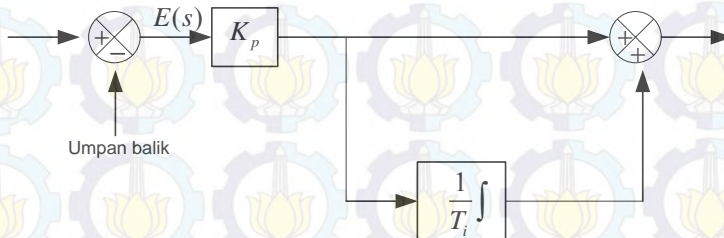
1. Keluaran kontroler integral membutuhkan suatu selang waktu yang menyebabkan kontroler ini membuat respon berjalan lebih lambat. Keluaran kontroler integral tetap bertahan pada nilai sebelumnya jika sinyal kesalahan berharga nol.

2. Keluaran kontroler integral akan menunjukkan kenaikan atau penurunan yang dipengaruhi besarnya sinyal kesalahan dan nilai  $K_i$  apabila harga sinyal kesalahan tidak nol.

3. Semakin besar nilai dari konstanta  $K_i$  maka mengakibatkan peningkatan osilasi sinyal keluaran kontroler integral, dan konstanta  $K_i$  yang berharga besar menyebabkan percepatan hilangnya *offset*. [26]

### 2.6.3. Kontroler Proporsional Integral

Kontroler proporsional ditambah integral (kontroler PI) adalah kontroler yang dapat saling melengkapi masing-masing kontroler P dan I dengan menggabungkan kedua kontroler secara paralel. Komponen-komponen dari kontroler P dan I masing-masing berfungsi untuk membuat reaksi dari sebuah sistem menjadi lebih cepat, menghasilkan perubahan awal yang besar, serta menghilangkan *offset*. [26]



**Gambar 2.5** Diagram blok kontroler PI

Jumlahan keluaran kontroler proporsional dan integral adalah keluaran kontroler PI. Kedua parameter P dan I sangat mempengaruhi karakteristik kontroler PI. Penentuan konstanta  $K_p$  dan  $T_i$  akan menimbulkan penonjolan sifat dari masing-masing elemen P dan I. Elemen kedua konstanta tersebut dapat dibuat lebih dominan dibanding yang lain. Kontribusi pengaruh pada respon sistem secara keseluruhan akan diberikan oleh konstanta yang dominan. [10]



## 2.7 PARTICLE SWARM OPTIMIZATION

*Particle Swarm Optimization* (PSO) terinspirasi dari perilaku kawanan hewan seperti ikan (*school of fish*), hewan herbivora (*herd*), dan burung (*flock*) yang kemudian tiap objek tersebut disederhanakan menjadi sebuah partikel. Suatu partikel dalam ruang memiliki posisi yang dikodekan sebagai vektor koordinat. Vektor posisi tersebut adalah keadaan yang sedang ditempati oleh suatu partikel di ruang pencarian dan dianggap sebagai keadaan yang sedang ditempati oleh suatu partikel di dalam ruang pencarian. Setiap partikel bergerak dengan kecepatan  $v$  [15].

*Particle Swarm Optimization* (PSO) menerapkan sifat masing-masing individu dalam satu kelompok besar dan menggabungkan berbagai sifat tersebut untuk menyelesaikan suatu permasalahan. *Particle Swarm Optimization* pertama kali muncul pada tahun 1995 dan pada saat itulah banyak dilakukan penelitian tentang metode PSO. Ciri khas metode ini adalah pengaturan kecepatan partikel secara heuristic dan probabilistik. Apabila suatu partikel memiliki kecepatan yang konstan maka jika jejak posisi partikel divisualisasikan maka akan membentuk garis lurus. Dengan adanya faktor eksternal yang menggerakkan partikel dalam ruang pencarian maka diharapkan partikel dapat mencapai titik optimal. Faktor eksternal tersebut antara lain yaitu posisi terbaik yang pernah dikunjungi suatu partikel, posisi terbaik seluruh partikel, dan faktor kreativitas yang berfungsi untuk melakukan eksplorasi [18].

*Particle Swarm Optimization* (PSO) memiliki kesamaan sifat seperti *Genetic Algorithm* (GA). Sistem ini diinisiasi oleh sebuah populasi solusi secara acak dan selanjutnya mencari titik optimum. Pada konteks optimisasi multivariabel, kawanan diasumsikan memiliki ukuran tertentu dengan setiap partikel posisi awalnya yang terletak pada lokasi yang acak dalam ruang multidimensi. Setiap partikel memiliki dua karakteristik yaitu kecepatan dan posisi. Setiap partikel bergerak dalam ruang tertentu dan mengingat posisi terbaik selama pencarian makanan. Setiap partikel menyampaikan posisi terbaiknya kepada partikel lain untuk ditempati juga oleh partikel lain tersebut [23].

Sebagai contoh, misalnya perilaku burung pada sekelompok kawanan burung. Meskipun setiap burung mempunyai keterbatasan kecerdasan, biasanya burung akan mengikuti kebiasaan seperti berikut :

1. Ketika berdiam diri, seorang burung tidak berada dekat dengan burung yang lain.

2. Burung mengarahkan terbangnya ke arah rata-rata keseluruhan burung.

3. Posisi burung rata-rata sama dengan rata-rata posisi burung yang lain untuk menjaga jarak antar burung agar tidak terlalu jauh dan tetap seimbang.

Dengan demikian perilaku kawanan burung akan berdasarkan pada kombinasi faktor kohesi (terbang bersama), separasi (jarak tidak terlalu dekat), dan penyesuaian (alignment) yaitu mengikuti arah bersama [18].

Jadi algoritma PSO dikembangkan berdasarkan model berikut ini :

1. Ketika seekor burung mendekati target atau makanan, maka burung tersebut akan memberitahu kawanan burung yang lainnya.
2. Secara tidak langsung, burung yang lain akan mengikuti arah tempat adanya makanan tersebut.
3. Ada memori pada setiap pikiran burung yaitu tentang apa yang sudah dilewati pada waktu sebelumnya.

Model ini disimulasikan pada ruang dan dimensi tertentu dengan sejumlah iterasi sehingga di setiap iterasi posisi partikel akan semakin mengarah ke target yang dituju yaitu fungsi minimal atau maksimal. Hal tersebut dilakukan hingga tercapai iterasi yang maksimum atau dapat juga digunakan kriteria penghentian lain [23].

Saat burung terbang mencari makanan, setiap burung menentukan posisi berdasarkan pengalamannya (nilai ini yang disebut *Pbest*) dan berdasarkan pengalaman dari partikel tetangganya (nilai ini yang disebut *Gbest*) [15]. Kecepatan partikel dan posisi partikel adalah dua istilah pada algoritma PSO pada saat terjadi proses pencarian makanan. Kedua istilah tersebut direpresentasikan pada persamaan :

$$v_i + c_1 r_1 (Pbest - x_i^k) + c_2 r_2 (Gbest - x_i^k) \quad (2.18)$$

$$x_i^{k+1} = x_i + v_i^{k+1} \quad (2.19)$$

Keterangan

*I* = Partikel ke-*i*

*k* = Iterasi ke-*k*

*C1* dan *C2* = *Learning Factor*

*Pbest* = Posisi terbaik yang dicapai partikel

*Gbest* = *Pbest* terbaik dalam populasi

$r_1$  dan  $r_2$

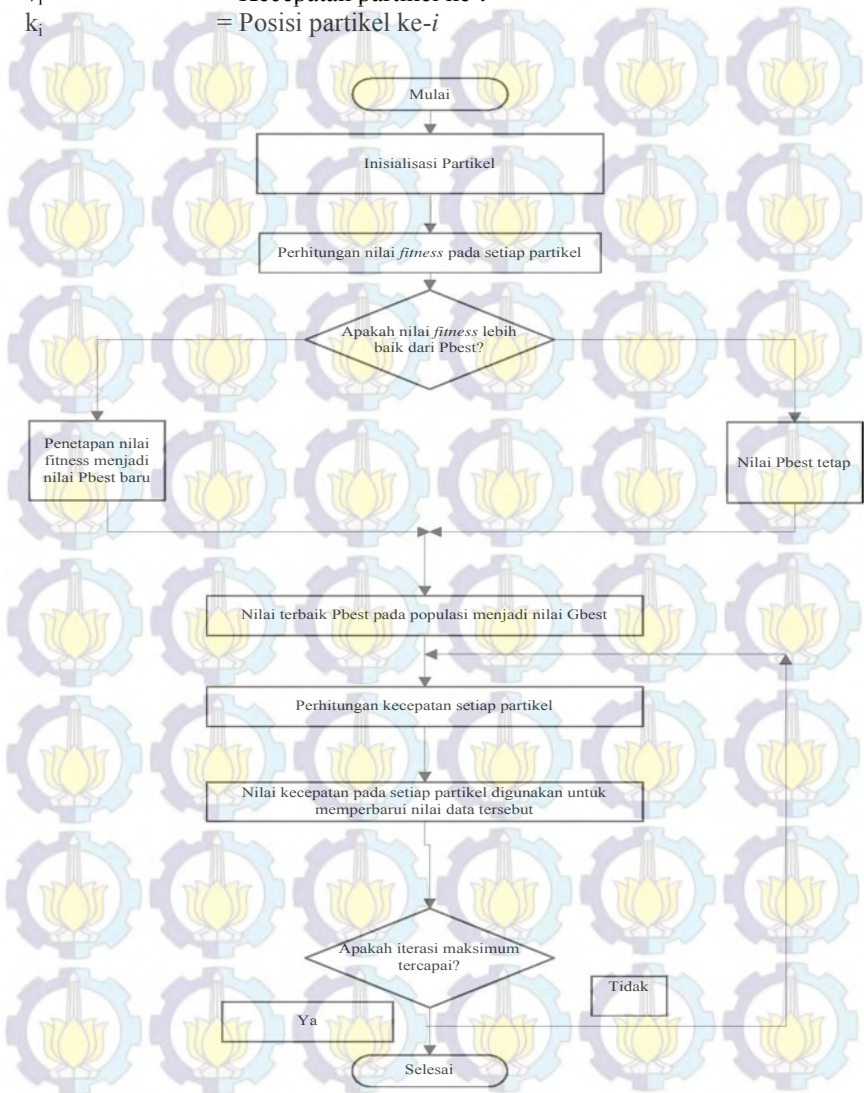
$v_i$

$k_i$

= Nilai random antara 0 dan 1

= Kecepatan partikel ke- $i$

= Posisi partikel ke- $i$



**Gambar 2.6** Flowchart koordinasi PSO



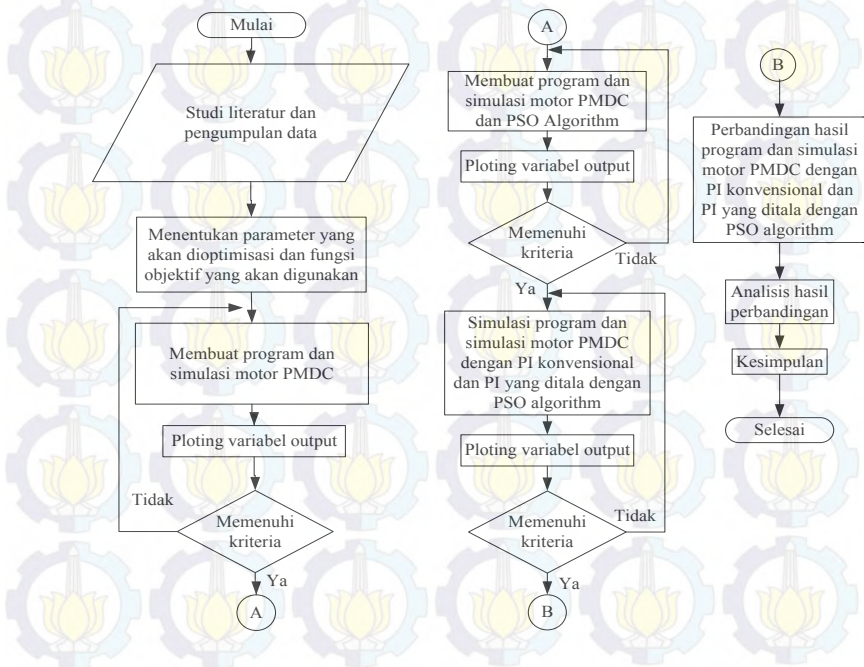
## **BAB 3**

### **PENERAPAN *PARTICLE SWARM OPTIMIZATION* PADA *OPTIMAL CONTROL PERMANENT MAGNET DIRECT CURRENT MOTOR***

Bab ini berisi tentang metode dan penyelesaian permasalahan pengaturan optimal motor DC magnet permanen dengan menggunakan PSO. Pada bab ini juga dijelaskan tentang langkah penyelesaian tugas akhir, parameter yang digunakan, dan langkah program PSO dalam pengaturan optimal motor DC magnet permanen.

#### **3.1. Algoritma Pengerjaan Tugas Akhir**

Flowchart pengerjaan Tugas Akhir ini ditunjukkan pada gambar 3.1.



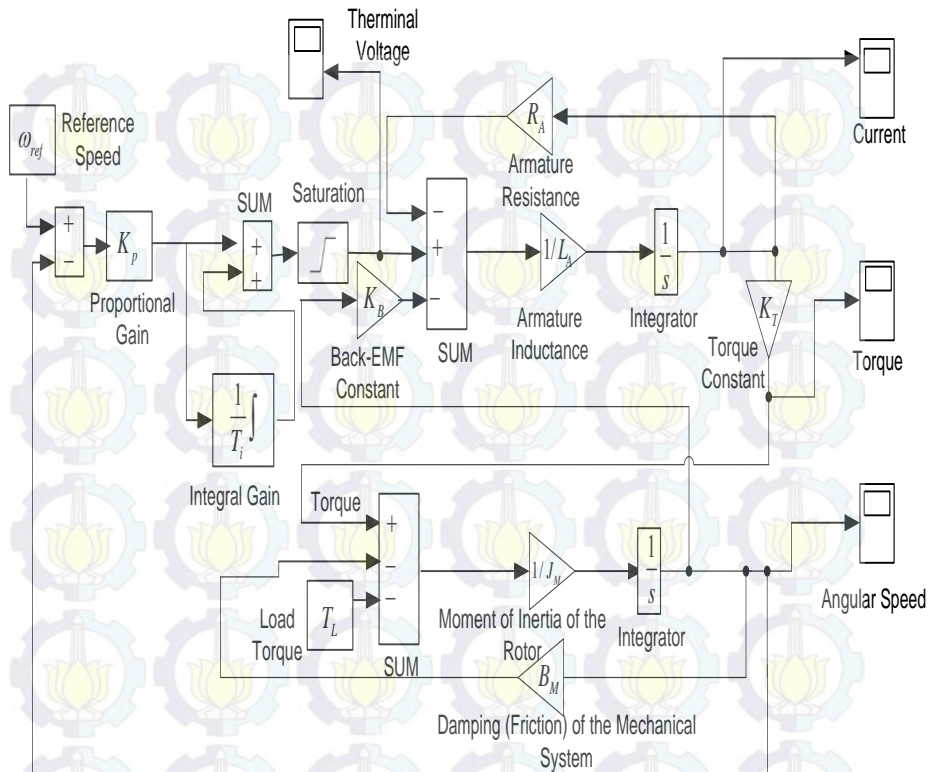
**Gambar 3.1.** Flowchart pengerjaan Tugas Akhir



Pengerjaan Tugas Akhir ini dimulai dengan melakukan studi literatur tentang motor DC magnet permanen terutama tentang pengaturan optimisasinya agar motor DC tersebut memperoleh suplai tegangan yang sesuai. Setelah dilakukan studi literatur maka selanjutnya dilakukan pengumpulan data dari berbagai macam paper dan jurnal ilmiah mengenai sistem motor DC magnet permanen tersebut. Setelah dilakukan pengumpulan data, langkah selanjutnya adalah menyusun program PSO dan skema motor DC magnet permanen yang dilengkapi kontroler PI. Kemudian dua program tersebut disimulasikan dalam sistem simulink motor DC magnet permanen. Sistem dibagi menjadi 2 perbandingan yaitu respon sistem dengan PI *conventional* dan respon sistem dengan PI yang dioptimisasi dengan PSO. Setelah simulasi selesai dilaksanakan maka dilakukan analisis tentang hasil simulasi tersebut. Setelah itu kemudian dibuat kesimpulan berdasarkan hasil analisis tersebut.

### **3.2. Sistem Motor DC Magnet Permanen dengan Kontroler PI**

Pada Tugas Akhir ini digunakan data sistem dari model skema motor DC magnet permanen yang dilengkapi dengan kontroler PI. Model sistem gabungan dapat dilihat pada Gambar 3.2, sedangkan parameter sistem ditunjukkan pada tabel 3.1.



**Gambar 3.2.** Skema Motor PMDC dengan Kontroler PI

### 3.2.1. Data Sistem [12]

Pada Tugas Akhir ini digunakan parameter sistem motor PMDC yang terdiri dari kecepatan sudut ( $\omega_m$ ), *Armature Resistance* ( $R_A$ ), *Armature Inductance* ( $L_A$ ), *moment inertia of the rotor* ( $J_M$ ), *damping (Friction) of the mechanical system* ( $B_M$ ), *Torque Constant* ( $K_T$ ), dan *Back-Emf Constant* ( $K_B$ ). Parameter sistem pada skema ini dapat dilihat pada tabel 3.1. berikut.

**Tabel 3.1.** Parameter Sistem [12]

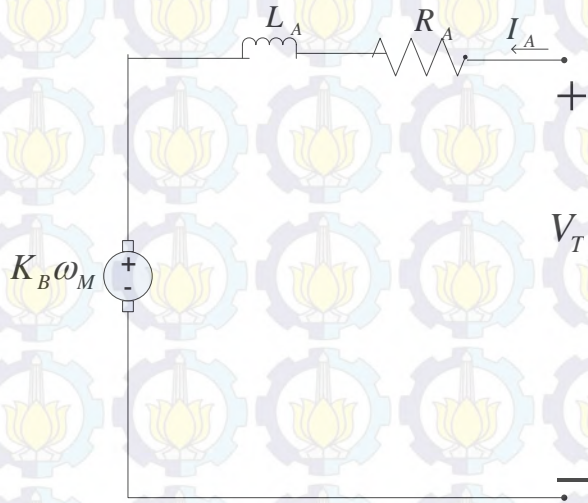
PMDC Parameters	Nilai
<i>Armature Resistance</i> (RA)	0,6 $\Omega$
<i>Armature Inductance</i> (LA)	0,012 H
<i>moment of inertia of the rotor</i> (JM)	0,0167 $\text{Kg m}^2/\text{s}^2$
<i>damping (Friction) of the mechanical system</i> (BM)	0,0167 Nms
<i>Torque Constant</i> (KT)	0.8 Nm/A
<i>Back-Emf Constant</i> (KB)	0.8 Vs/rad

Dalam simulasi sistem motor DC magnet permanen tersebut, kecepatan motor PMDC bergantung pada tegangan input yang dikontrol oleh kontroler PI agar kecepatannya tetap sesuai dengan kecepatan referensinya meskipun motor PMDC tersebut berada pada kondisi pembebanan yang berbeda-beda. Dengan kondisi pembebanan yang diberikan pada motor PMDC tersebut diharapkan motor PMDC tetap berada pada performansi yang optimal [16]

*Error* yang dikontrol oleh kontroler PI digunakan untuk mengamati kecepatan motor PMDC agar kecepatannya tetap optimal pada kondisi kecepatan motor yang tidak sesuai dengan kecepatan referensinya serta pada kondisi pembebanan yang diberikan kepada sistem motor PMDC. Selain itu untuk mengamati performansi sistem motor PMDC adalah dengan mengamati nilai *overshoot* dan *settling time* pada respon sistem motor PMDC [17].

### 3.3. Pemodelan Linear Motor PMDC [13]

Persamaan linear motor dc dapat diturunkan dengan menggunakan hukum-hukum dasar rangkaian listrik serta prinsip-prinsip hubungan elektromekanik melalui rangkaian ekivalen motor dc sehingga persamaan linear tersebut dapat digunakan untuk analisis dan perencanaan teknik pengaturan motor. Pemodelan linear motor dc terdiri dari persamaan elektrik motor dan persamaan mekanik motor.



Gambar 3.3. Rangkaian Ekivalen Motor PMDC

#### 3.3.1. Persamaan Elektrik Motor DC [13]

Persamaan elektrik motor DC berdasarkan hukum Newton yang dikombinasikan dengan hukum Kirchoff dapat didefinisikan sebagai berikut :

$$V_T(t) = E_B(t) + R_A I_A(t) + L_A \frac{dI_A(t)}{dt} \quad (3.1)$$

$$E_B(t) = K_B \phi \omega_M(t) \quad (3.2)$$

karena motor yang digunakan adalah motor PMDC, maka nilai fluks magnet menjadi konstan.



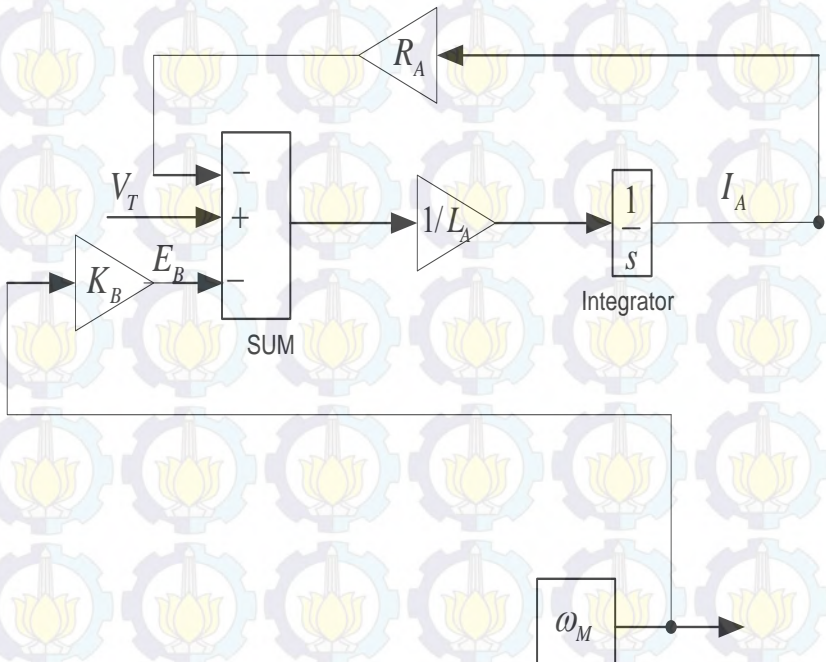
$$E_B(t) = K_B \omega_M(t) \quad (3.3)$$

$$V_T(t) = K_B \omega_M(t) + R_A I_A(t) + L_A \frac{dI_A(t)}{dt} \quad (3.4)$$

$$L_A \frac{dI_A(t)}{dt} = V_T(t) - K_B \omega_M(t) - R_A I_A(t) \quad (3.5)$$

$$\frac{dI_A(t)}{dt} = \frac{V_T(t)}{L_A} - \frac{K_B \omega_M(t)}{L_A} - \frac{R_A I_A(t)}{L_A} \quad (3.6)$$

Dari hasil persamaan elektrik motor PMDC tersebut, maka dapat dirancang pemodelan motor PMDC sebagai berikut :



**Gambar 3.4.** Pemodelan Motor PMDC Melalui Persamaan Elektrik

$I_A$	: arus jangkar
$V_T$	: tegangan terminal
$E_B$	: <i>back emf</i>
$K_B$	: konstanta <i>back-emf</i>
$\omega_M$	: kecepatan motor
$R_A$	: resistansi jangkar
$I_A$	: arus jangkar
$L_A$	: induktansi jangkar

### 3.3.2. Persamaan Mekanik Motor DC [13]

Persamaan mekanik motor DC berdasarkan hukum Newton yang dikombinasikan dengan hukum Kirchoff dapat didefinisikan sebagai berikut :

$$T_E(t) = J_M \frac{d\omega_M(t)}{dt} + T_L(t) + B_M \omega_M(t) \quad (3.7)$$

$$T_E(t) = K_T \phi I_A(t) \quad (3.8)$$

karena motor yang digunakan adalah motor PMDC, maka nilai fluks magnet menjadi konstan.

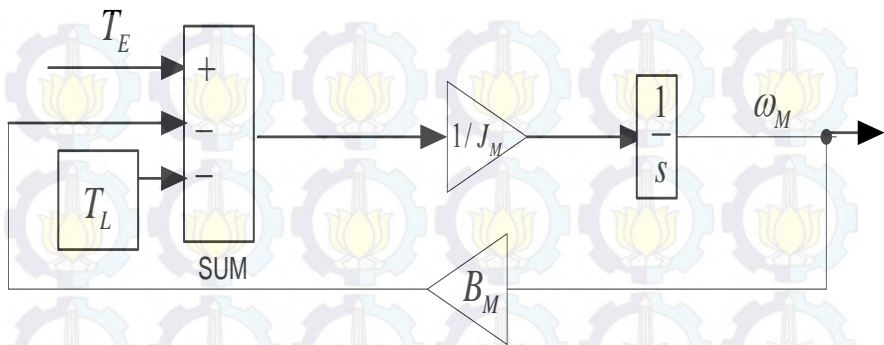
$$T_E(t) = K_T I_A(t) \quad (3.9)$$

$$K_T I_A(t) = J_M \frac{d\omega_M(t)}{dt} + T_L(t) + B_M \omega_M(t) \quad (3.10)$$

$$J_M \frac{d\omega_M(t)}{dt} = K_T I_A(t) - B_M \omega_M(t) - T_L(t) \quad (3.11)$$

$$\frac{d\omega_M(t)}{dt} = \frac{K_T I_A(t)}{J_M} - \frac{B_M \omega_M(t)}{J_M} - \frac{T_L(t)}{J_M} \quad (3.12)$$

Dari hasil persamaan mekanik motor PMDC tersebut, maka dapat dirancang pemodelan motor PMDC sebagai berikut :



**Gambar 3.5.** Pemodelan Motor PMDC Melalui Persamaan Mekanik

$\omega_M$  : kecepatan motor

$K_T$  : konstanta torsi

$I_A$  : arus jangkar

$B_M$  : koefisien redaman sistem mekanik

$T_L$  : torsi beban

$J_M$  : inersia rotor

### 3.3.3 Sistem dengan Penambahan Kontroler PI

Aplikasi kontroler PI dilakukan dengan penambahan kontroler PI pada sistem. Aksi kontrol PI adalah untuk mengatasi gangguan beban berupa torsi mekanik. Sinyal kontrol dari PI yang berupa tegangan diinputkan ke sistem motor PMDC [7].

Percobaan dilakukan pada sistem dengan melakukan penambahan kontroler PI. Gangguan yang terjadi pada sistem motor PMDC adalah torsi mekanik. Jika terjadi perubahan beban pada sistem maka sistem akan mengalami gangguan karena disebabkan sistem pada motor PMDC tersebut saling terinterkoneksi menjadi satu. Gangguan-gangguan yang terjadi karena perubahan beban tersebut adalah torsi mekanik. Respon yang diamati yaitu berupa kecepatan sudut rotor ( $\omega_M$ ), arus jangkar motor ( $I_A$ ), dan torsi elektrik motor ( $T_E$ ) berdasarkan input tegangan ( $V_T$ ) pada sistem motor PMDC. Kontroler PI digunakan untuk mengontrol input tegangan ( $V_T$ ) sehingga kecepatan

sudut rotor ( $\omega_m$ ) motor PMDC yang diperoleh sesuai dengan kecepatan sudut yang diharapkan dan membuat respon sistem menjadi lebih optimal.

Perencanaan parameter kontroler PI dapat dijabarkan sebagai berikut. Dengan menggunakan model matematika sistem elektromekanika maka diperoleh model matematika (fungsi alih) dari motor DC dengan penguatan medan konstan yaitu :

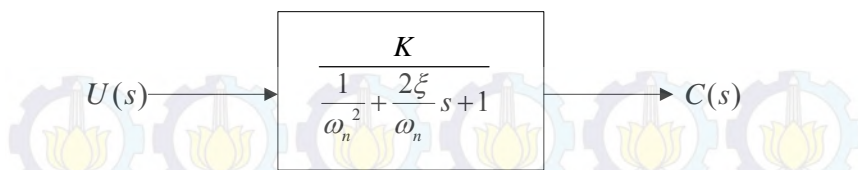
$$G(s) = \frac{\omega(s)}{V_{in}(s)} = \frac{K_T}{(L_A \cdot s + R_A)(J_M s + B_M) + (K_T \cdot K_B)} \quad (3.13)$$

Dari parameter motor PMDC diketahui bahwa  $K_T = 0,8$ ;  $L_A = 0,012$ ;  $R_A = 0,6$ ;  $J_M = 0,0167$ ;  $B_M = 0,0167$  ;  $K_B = 0,8$ ; dengan memasukkan parameter motor PMDC didapatkan :

$$\begin{aligned} \frac{\omega(s)}{V_{in}(s)} &= \frac{0,8}{(0,012s + 0,6)(0,0167s + 0,016) + (0,8)(0,8)} \\ &= \frac{0,8}{0,0002s^2 + 0,00019s + 0,01002s + 0,0096 + 0,64} \\ &= \frac{0,8}{0,0002s^2 + 0,01021s + 0,6496} \\ \frac{\omega(s)}{V_{in}(s)} &= \frac{K}{\frac{1}{\omega_n^2} s^2 + \frac{2\xi}{\omega_n} s + 1} \quad (3.14) \\ n(s) &= \frac{1,23153}{0,00031s^2 + 0,01572s + 1} \end{aligned}$$

Suatu plant orde kedua tanpa delay dapat direpresentasikan dalam bentuk diagram blok seperti berikut ini :

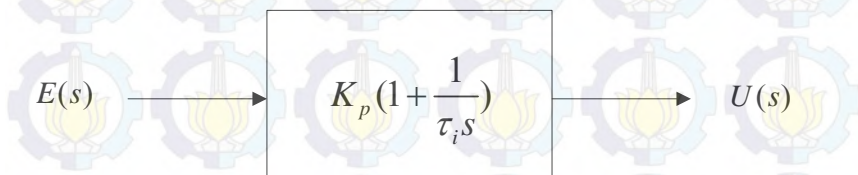




**Gambar 3.6.** Diagram Blok Plant Orde Kedua Tanpa Delay

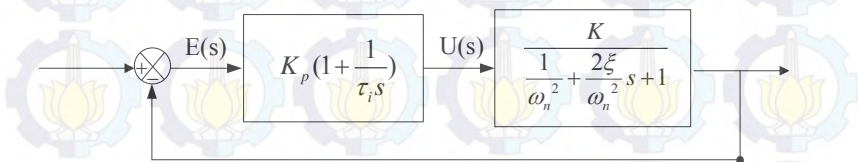
$K$  adalah Gain overall,  $\omega_n$  adalah frekuensi alami tak teredam, dan  $\xi$  adalah rasio peredaman.

Sedangkan diagram blok dari kontroler PI adalah sebagai berikut :



**Gambar 3.7.** Diagram Blok Kontroler PI

$K_p$  adalah penguatan proporsional dan  $\tau_i$  adalah waktu integral. Jika kontroler proporsional ditambah integral diterapkan pada plant orde kedua tanpa delay dalam suatu sistem pengaturan maka diagram bloknnya adalah sebagai berikut :



**Gambar 3.8.** Diagram Blok Kontroler PI yang Diterapkan Pada Plant Orde Kedua Tanpa Delay

$$n(s) = \frac{1,23153}{0,00031s^2 + 0,01572s + 1}$$

Oleh karena itu berdasarkan persamaan (3.14), perencanaan parameter kontroler PI dapat dijabarkan sebagai berikut :

Dari persamaan tersebut maka diperoleh nilai  $K = 1,23153$  sedangkan nilai  $\omega_n$  diperoleh dari persamaan :

$$\begin{aligned} \frac{1}{\omega_n^2} &= 0,00031 & (3.15) \\ \omega_n &= \sqrt{\frac{1}{0,00031}} \\ \omega_n &= 56,79618 \end{aligned}$$

Nilai  $\xi$  dapat diperoleh dari persamaan berikut ini :

$$\begin{aligned} \frac{2\xi}{\omega_n} &= 0,01572 & (3.16) \\ \xi &= \frac{(0,01572)(56,79618)}{2} \\ \xi &= 0,44642 \end{aligned}$$

Besarnya waktu integral ( $\tau_i$ ) dapat diperoleh dari persamaan berikut ini :

$$\begin{aligned} \tau_i &= \frac{2\xi}{\omega_n} & (3.17) \\ &= \frac{(2)(0,44642)}{56,79618} \\ \tau_i &= 0,01572 \end{aligned}$$

Besarnya penguatan proporsional ( $K_p$ ) dapat diperoleh dari persamaan berikut ini :

$$K_p = \frac{2\xi}{\tau\omega_n K} \quad (3.18)$$

$$= \frac{(2)(0,44642)}{(0,1)(56,79618)(1,23153)}$$

$$K_p = 0,12765$$

Besarnya penguatan integral ( $K_i$ ) dapat diperoleh dari persamaan berikut ini :

$$K_i = \frac{K_p}{T_i} \quad (3.19)$$

$$= \frac{0,12765}{0,01572}$$

$$K_i = 8.12023$$

Dari hasil perhitungan diatas, maka secara keseluruhan diperoleh nilai-nilai sebagai berikut :

$$K = 1,23153$$

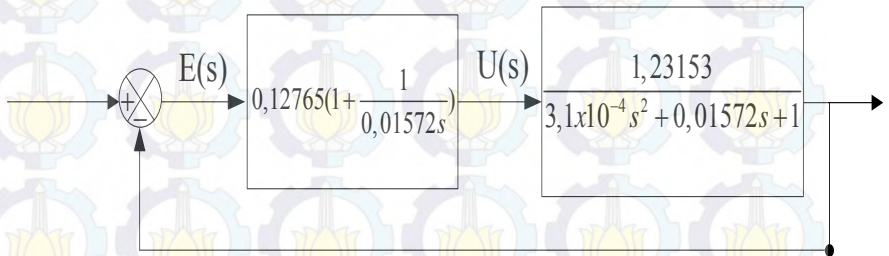
$$\omega_n = 56,79618$$

$$\xi = 0,44642$$

$$\tau_i = 0,01572$$

$$K_p = 0,12765$$

Diagram bloknya dalam suatu sistem pengaturan adalah sebagai berikut :



**Gambar 3.9.** Diagram Blok Sistem Pengaturan Motor PMDC

Dari hasil perhitungan diatas maka diperoleh nilai  $K_p = 0,12765$ ,  $\tau_i = 0,01572$  sehingga diperoleh pendekatan nilai-nilai parameter kontroler PI adalah sebagai berikut :

**Tabel 3.2** Parameter Kontroler PI

$K_p$	$\tau_i$
$0,1 \leq K_p \leq 0,13$	$0,01 \leq \tau_i \leq 0,03$

Setelah kontroler PI dipasang pada sistem, langkah selanjutnya adalah mengamati respon sistem. Respon sistem yang diamati yaitu kecepatan sudut rotor ( $\omega_M$ ), arus jangkar motor ( $I_A$ ), dan torsi elektrik motor ( $T_E$ ) berdasarkan input berupa tegangan ( $V_T$ ). Untuk mendapatkan respon sistem yang baik maka diperlukan koordinasi kontroler PI yang optimal.

### 3.3.4 Pemodelan Gabungan Skema Penggerak Motor PMDC dengan Kontroler PI [1]

Torsi motor PMDC ( $T$ ), berkaitan dengan arus jangkar ( $I_A$ ) dengan faktor konstan  $K_T$  :  $T = K_T I_A$ . Persamaan *back emf* ( $E_B$ ) berkaitan dengan kecepatan rotasi yaitu  $E_B = K_B \omega_M$ , dalam unit SI  $K_T$  (konstanta torsi) sama dengan  $K_B$  (konstanta *back-emf*). Persamaan motor PMDC berdasarkan hukum Newton dikombinasikan dengan hukum Kirchoff :

$$J_M \frac{d\omega_M}{dt} + B_M \omega_M = K_T I_A - T_L \quad (3.20)$$

$$L_A \frac{dI_A}{dt} + R_A I_A = V_T - K_B \omega_M \quad (3.21)$$

$J_M$  adalah momen inersia,  $B_M$  adalah koefisien redaman pada sistem mekanik,  $R_A$  adalah resistansi jangkar, dan  $L_A$  adalah induktansi jangkar. Persamaan diatas dapat dinyatakan dengan memilih kecepatan rotasi dan arus listrik sebagai variabel keadaan dan tegangan sebagai input. Output dipilih untuk menjadi kecepatan rotasi [11].



$$\dot{x} = Ax + Bu$$

$$y = Cx + Du$$

$$\frac{d}{dt} \begin{bmatrix} \omega_M \\ I_A \end{bmatrix} = \begin{bmatrix} -\frac{B_M}{J_M} & \frac{K_T}{J_M} \\ -\frac{K_B}{L_A} & -\frac{R_A}{L_A} \end{bmatrix} \begin{bmatrix} \omega_M \\ I_A \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{L_A} \end{bmatrix} V_T + \begin{bmatrix} -\frac{T_L}{J_M} \\ 0 \end{bmatrix} \quad (3.22)$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \omega_M \\ I_A \end{bmatrix} \quad (3.23)$$

Dalam domain waktu, kontroler PI direpresentasikan dalam persamaan berikut ini :

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt \quad (3.24)$$

$e(t)$  adalah sistem *error* yang dipilih,  $u(t)$  sebagai variabel control,  $K_p$  sebagai *proportional gain* dan  $K_i$  adalah *integral gain*. Terdapat beberapa penambahan karakteristik khusus pada setiap koefisien pada kontroler PI untuk respon output dari sistem. Oleh sebab itu, pemilihan *gain* adalah hal penting yang dilakukan dalam praktik kontroler ini [10].

Gambar 3.2 merupakan gabungan skema penggerak motor PMDC dengan kontroler PI. Sistem penggerak motor PMDC menggunakan penyetelan optimisasi sekumpulan partikel PSO. Model kontrol motor PMDC secara umum yang dikontrol oleh kontroler PI divalidasi untuk mencapai kecepatan referensi yang diharapkan pada berbagai kondisi pembebanan yang berbeda-beda yang diberikan pada sistem motor PMDC. Dengan kondisi pembebanan yang berbeda-beda pada motor PMDC maka diharapkan performansi sistem motor PMDC tetap berada pada kondisi yang optimal [3].

*Error* yang dikontrol oleh kontroler PI divalidasi untuk melacak kecepatan referensi motor PMDC agar tetap optimal pada kondisi kecepatan motor yang tidak sesuai dengan kecepatan referensinya serta pada kondisi perubahan parameter beban yang terjadi

respons sistem motor PMDC [6]. Gambar schematic diagram dari motor PMDC ditunjukkan pada gambar 3.10.

**Gambar 3.10.** Diagram Skematik Motor PMDC

Metode yang digunakan untuk menyelesaikan permasalahan pada tugas akhir ini adalah metode algoritma PSO. Algoritma PSO digunakan untuk mencari kombinasi dari parameter-parameter yang paling optimal. PSO terdiri dari sejumlah partikel yang akan dioptimisasi atau berapa kali partikel tersebut dioptimisasi. Dalam tugas akhir ini parameter yang akan dioptimisasi pada kontroler PI ada 2, yaitu  $K_p$  dan  $T_i$ . Nilai-nilai pada  $K_p$  dan  $T_i$  diasumsikan sebagai dimensi pada PSO yang akan menjadi koordinat lokasi vektor pada populasi PSO. PSO akan mengoptimisasi nilai  $K_p$  dan  $T_i$  hingga iterasi yang telah ditentukan. Kombinasi terbaik diperoleh melalui evaluasi setiap kombinasi dari nilai  $K_p$  dan  $T_i$ . Kombinasi terbaik dari proses optimisasi tersebut akan menghasilkan sistem yang memiliki osilasi

dengan *overshoot* terkecil dan *settling time* tercepat. Evaluasi pada sistem dilakukan dengan menghitung nilai fungsi objektif dari respon sistem. Pada PSO fungsi objektif adalah fungsi evaluasi dari masing-masing populasi. Dalam tugas akhir ini, fungsi objektif yang digunakan untuk menentukan fungsi evaluasi adalah *Integral of Time Multiplied Absolute Error* (ITAE). Fungsi evaluasi ITAE ini akan menghitung luasan dari respon yang ditampilkan. Jika penjumlahan itu memiliki nilai yang paling kecil maka dianggap memiliki respon paling baik dengan *overshoot* dan *settling time* yang baik. ITAE ditunjukkan oleh persamaan,

$$ITAE = \int_0^t t | \Delta \omega(t) | dt \quad (3.25)$$

Parameter PSO yang digunakan pada tugas akhir ini ditunjukkan pada tabel 3.3.

**Tabel 3.3** Parameter PSO

Jumlah swarm	50
Jumlah variabel	2
Iterasi maksimal	20
<i>C1</i>	2
<i>C2</i>	2
<i>Inertia</i> (w)	0.4-0.9

Iterasi maksimal adalah banyaknya proses pengulangan yang dilakukan. *Social* dan *Cognitive constant* merupakan konstanta yang ditetapkan pada metode optimisasi dengan menggunakan algoritma PSO.

Simulasi dilakukan dengan 20 kali iterasi. Tabel 3.4 menunjukkan parameter PI yang telah dioptimisasi dengan PSO. Nilai-nilai tersebut diperoleh dari hasil simulasi. Hasil optimisasi parameter PI menggunakan PSO menghasilkan nilai *K<sub>p</sub>* sebesar 0,1247 dan nilai *T<sub>i</sub>* sebesar 0,0281. Nilai *K<sub>p</sub>* dan *T<sub>i</sub>* tersebut menunjukkan bahwa populasi telah menemukan koordinasi dimensi terbaik pada nilai tersebut melalui nilai tersebut. Nilai-nilai pada *K<sub>p</sub>* dan *T<sub>i</sub>* diasumsikan sebagai dimensi pada PSO yang akan menjadi koordinat lokasi vektor pada populasi PSO.

**Tabel 3.4** Hasil optimisasi parameter PI menggunakan PSO

$K_p$	$\tau_i$
0,1247	0,0281



## **BAB 4**

### **SIMULASI DAN ANALISIS**

Pada bab ini akan dibahas tentang simulasi yang dihasilkan oleh program PSO untuk menyelesaikan permasalahan pengaturan optimal motor PMDC. Simulasi dalam Tugas Akhir ini digunakan untuk mencari koordinasi parameter kontroler PI dengan menggunakan *Particle Swarm Optimization* (PSO). Selain itu, Tugas Akhir ini juga digunakan untuk mengetahui karakteristik plant motor PMDC sehingga dapat digunakan untuk menganalisis peningkatan performa motor PMDC. Aksi kontrol PI adalah untuk mengatasi gangguan beban berupa torsi mekanik atau torsi beban. Untuk mengatasi gangguan beban pada motor PMDC maka dilakukan pengaturan *gain* kontroler PI. Untuk mempermudah analisis maka sistem dimodelkan dalam bentuk *laplace*, karena dengan menggunakan *laplace* maka akan lebih mudah dalam mendeteksi sifat dan karakteristik motor.

Dalam simulasi sistem motor dc magnet permanen diberi kasus gangguan yaitu berupa torsi mekanik. Simulasi dilakukan untuk menganalisis respon sistem motor PMDC. Respon sistem yang dianalisis adalah kecepatan sudut rotor ( $\omega_m$ ), arus jangkar motor ( $I_a$ ), torsi elektrik motor ( $T_e$ ), dan tegangan ( $V_t$ ). Batas tegangan maksimal yang dibatasi oleh *signal limiter* adalah 103,35 V (6% dari tegangan 97,5 V saat  $\omega_{ref} = 120$  rad/sec). Sistem dibagi menjadi 2 perbandingan yaitu respon sistem dengan PI *conventional* dan respon sistem dengan PI yang dioptimisasi dengan PSO. Pemilihan sistem PI dilakukan dengan data dari literatur, sedangkan parameter PSO juga diperoleh dari literatur.

#### **4.1. Parameter PSO Pada Sistem Motor PMDC**

Parameter yang digunakan pada PSO adalah jumlah partikel, jumlah iterasi maksimal, jumlah variabel, konstanta sosial, konstanta kognitif, dan bobot inersia. Nilai-nilai pada parameter tersebut tercantum pada tabel 4.1. Parameter tersebut digunakan untuk melakukan simulasi pada pengaturan optimal motor PMDC. Sistem yang dipakai adalah sistem motor PMDC. Algoritma PSO digunakan untuk mencari koordinasi kontroler PI.

**Tabel 4.1** Parameter PSO

Parameter	Nilai
Jumlah partikel	50
Jumlah iterasi maksimal	20
Jumlah variabel	2
Wmax	0.9
Wmin	0.4
C1	2
C2	2

Pada tugas akhir ini dilakukan simulasi pada program sistem motor PMDC menggunakan algoritma PSO. Evaluasi pada sistem dilakukan dengan menghitung nilai fungsi objektif dari respon sistem. Pada PSO fungsi objektif adalah fungsi evaluasi dari masing-masing populasi. Dalam tugas akhir ini, fungsi objektif yang digunakan untuk menentukan fungsi evaluasi adalah *Integral of Time Multiplied Absolute Error* (ITAE). Fungsi evaluasi ITAE ini akan menghitung luasan dari respon yang ditampilkan. Jika penjumlahan itu memiliki nilai yang paling kecil maka dianggap memiliki respon paling baik dengan *overshoot* dan *settling time* yang baik.

Dari hasil simulasi pada tugas akhir ini diperoleh dua kasus yang berbeda. Simulasi pertama yaitu mengamati respon yang terjadi bila sistem disimulasikan dengan menggunakan kontroler PI yang ditala secara konvensional. Simulasi kedua yaitu mengamati respon yang terjadi bila sistem disimulasikan dengan menggunakan kontroler PI yang dioptimisasi dengan menggunakan PSO.

#### 4.2 Respon Sistem dengan PI dan PI yang dioptimisasi dengan PSO

Gangguan pada sistem motor PMDC pada tugas akhir ini adalah torsi mekanik atau torsi beban. Pada jangka waktu 0-25 detik, tidak ada gangguan berupa torsi mekanik pada sistem motor PMDC, sedangkan pada jangka waktu 25-50 detik terdapat gangguan berupa torsi mekanik. Torsi mekanik yang diberikan pada sistem dibatasi hingga 2 Nm dan *set point* kecepatan motor PMDC adalah 100 rad/sec dan dibatasi hingga 120 rad/sec. Pada tugas akhir ini dilakukan 2 studi kasus. Studi kasus pertama yaitu dilakukan pengamatan respon sistem motor PMDC pada saat kecepatan referensi motor PMDC adalah 100 rad/sec dan motor diberi gangguan torsi mekanik sebesar 1 Nm dan 2

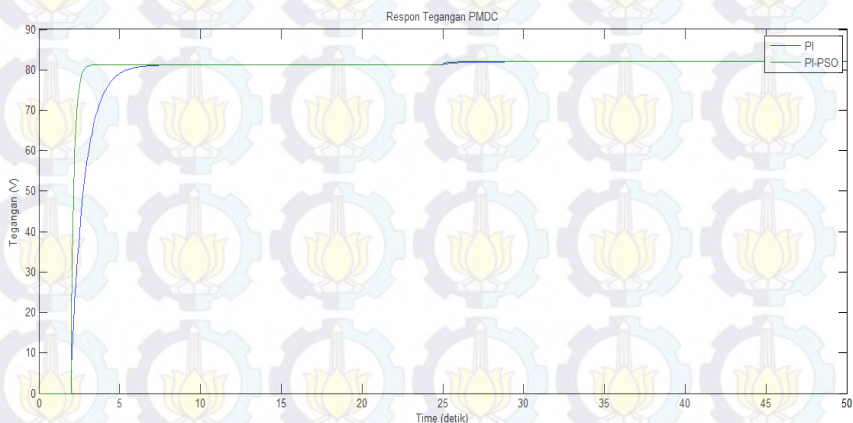
Nm. Studi kasus kedua yaitu dilakukan pengamatan respon sistem motor PMDC pada saat kecepatan referensi motor PMDC adalah 120 rad/sec dan motor diberi gangguan torsi mekanik sebesar 1 Nm dan 2 Nm.

#### 4.2.1 Respon Sistem dengan Kecepatan Referensi 100 rad/sec

##### 4.2.1.1 Respon Tegangan ( $V_T$ ) Sistem Motor PMDC

Studi kasus yang pertama adalah mengamati respon tegangan sistem motor PMDC dengan gangguan torsi mekanik 1 Nm, sedangkan studi kasus yang kedua adalah mengamati respon tegangan sistem motor PMDC dengan gangguan torsi mekanik 2 Nm. Respon tegangan sistem motor PMDC dengan gangguan torsi mekanik 1 Nm ditunjukkan pada gambar 4.1. Pengamatan dilakukan pada detik ke-0 hingga detik ke-50 sehingga diperoleh respon tegangan pada sistem pada kondisi motor tidak diberi gangguan torsi mekanik maupun kondisi pada saat motor diberi gangguan torsi mekanik.

Gambar 4.4 adalah gambar respon tegangan motor PMDC pada jangka waktu 0-50 detik. Motor PMDC mulai beroperasi pada detik ke-2. Garis biru pada gambar tersebut menunjukkan respon sistem yang dikontrol dengan kontroler PI dengan metode konvensional, sedangkan garis hijau pada gambar tersebut menunjukkan respon sistem yang dikontrol dengan kontroler PI yang dioptimisasi oleh algoritma PSO.



**Gambar 4.1** Respon Tegangan Motor PMDC dengan Gangguan Torsi Mekanik 1 Nm ( $\omega_{ref} = 100$  rad/sec)



#### 4.2.1.1.1 Respon Tegangan Motor Saat Sistem tidak diberi Gangguan Beban ( $T_L = 0 \text{ Nm}$ , $\omega_{\text{ref}} = 100 \text{ rad/sec}$ )

Pada saat motor tidak diberi gangguan beban nilai tegangan motor PMDC adalah 81,25 V. Perbandingan *overshoot* dan *settling time* respon tegangan motor PMDC pada jangka waktu 0-25 detik (tidak diberi gangguan beban) dapat dilihat pada tabel 4.2.

**Tabel 4.2** Perbandingan *overshoot* dan *settling time* respon tegangan motor PMDC pada 0-25 detik ( $\omega_{\text{ref}} = 100 \text{ rad/sec}$ )

Metode Kontrol	PI	PI PSO
<i>Overshoot</i> (V)	0	0
<i>Settling time</i> (det)	9,874	3,847

Dapat dilihat pada tabel 4.2 bahwa sistem dengan kontroler PI pada jangka waktu 0-25 detik tidak memiliki nilai *overshoot* dan memiliki nilai *settling time* sebesar 9,874 detik, sedangkan sistem dengan kontroler PI yang dioptimisasi dengan menggunakan PSO tidak memiliki nilai *overshoot* dan memiliki nilai *settling time* sebesar 3,847 detik. Oleh karena itu dapat disimpulkan bahwa pada periode 0-25 detik (pada saat sistem tidak diberi gangguan beban) respon sistem dengan kontroler PI yang dioptimisasi dengan PSO lebih baik dibandingkan respon sistem dengan kontroler PI konvensional karena respon sistem kembali ke kondisi *steady state* lebih cepat.

#### 4.2.1.1.2 Respon Tegangan Motor Saat Sistem diberi Gangguan Beban ( $T_L = 1 \text{ Nm}$ , $\omega_{\text{ref}} = 100 \text{ rad/sec}$ )

Pada jangka waktu 25-50 detik terdapat gangguan berupa torsi mekanik sebesar 1 Nm. Ketika motor diberi gangguan berupa beban, maka kecepatannya akan turun. Oleh karena itu tegangan input pada motor harus dinaikkan agar kecepatan motor dapat kembali ke kecepatan yang diinginkan karena salah satu cara dalam pengaturan kecepatan motor PMDC adalah dengan mengontrol tegangan inputnya. Oleh karena itu saat motor diberi beban torsi mekanik 1 Nm nilai tegangannya naik menjadi 82 V sehingga pada waktu 25-50 detik nilai tegangan motor menjadi 82 V. Perbandingan *overshoot* dan *settling time* respon tegangan motor PMDC pada jangka waktu 25-50 detik (diberi gangguan beban) dapat dilihat pada tabel 4.3.



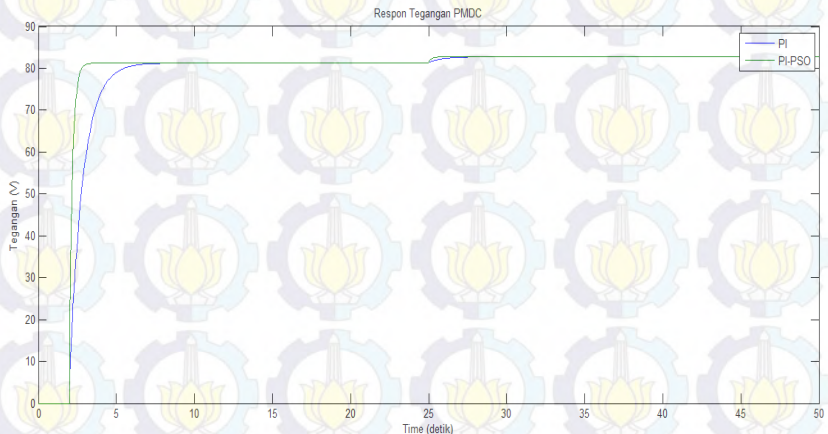
**Tabel 4.3** Perbandingan *overshoot* dan *settling time* respon tegangan motor PMDC saat 25-50 detik ( $T_L = 1 \text{ Nm}$ ,  $\omega_{\text{ref}} = 100 \text{ rad/sec}$ )

Metode Kontrol	PI	PI PSO
<i>Overshoot</i> (V)	0	0
<i>Settling time</i> (det)	28,87	25,94

Dapat dilihat pada tabel 4.3 bahwa sistem dengan kontroler PI pada jangka waktu 25-50 detik memiliki *overshoot* sebesar 0 V dan memiliki nilai *settling time* sebesar 28,87 detik, sedangkan sistem dengan kontroler PI yang dioptimisasi dengan menggunakan PSO tidak memiliki nilai *overshoot* dan memiliki nilai *settling time* sebesar 25,94 detik. Oleh karena itu dapat disimpulkan bahwa pada periode 25-50 detik (pada saat sistem diberi gangguan beban) respon sistem dengan kontroler PI yang dioptimisasi dengan PSO lebih baik dibandingkan respon sistem dengan kontroler PI konvensional.

#### 4.2.1.1.2 Respon Tegangan Motor Saat Sistem diberi Gangguan Beban ( $T_L = 2 \text{ Nm}$ , $\omega_{\text{ref}} = 100 \text{ rad/sec}$ )

Respon tegangan sistem motor PMDC dengan gangguan torsi mekanik 2 Nm ditunjukkan pada gambar 4.2.



**Gambar 4.2** Respon Tegangan Motor PMDC dengan Gangguan Torsi Mekanik 2 Nm ( $\omega_{\text{ref}} = 100 \text{ rad/sec}$ )

Pada jangka waktu 25-50 detik terdapat gangguan berupa torsi mekanik sebesar 2 Nm. Ketika motor diberi gangguan berupa beban, maka kecepatannya akan turun. Oleh karena itu tegangan input pada motor harus dinaikkan agar kecepatan motor dapat kembali ke kecepatan yang diinginkan karena salah satu cara dalam pengaturan kecepatan motor PMDC adalah dengan mengontrol tegangan inputnya. Oleh karena itu saat motor diberi beban torsi mekanik 2 Nm nilai tegangannya naik menjadi 82,75 V sehingga pada waktu 25-50 detik nilai tegangan motor menjadi 82,75 V. Perbandingan *overshoot* dan *settling time* respon tegangan motor PMDC pada jangka waktu 25-50 detik (diberi gangguan beban) dapat dilihat pada tabel 4.4.

**Tabel 4.4** Perbandingan *overshoot* dan *settling time* respon tegangan motor PMDC saat 25-50 detik ( $T_L = 2 \text{ Nm}$ ,  $\omega_{ref} = 100 \text{ rad/sec}$ )

Metode Kontrol	PI	PI PSO
<i>Overshoot</i> (V)	0	0
<i>Settling time</i> (det)	29,45	26,05

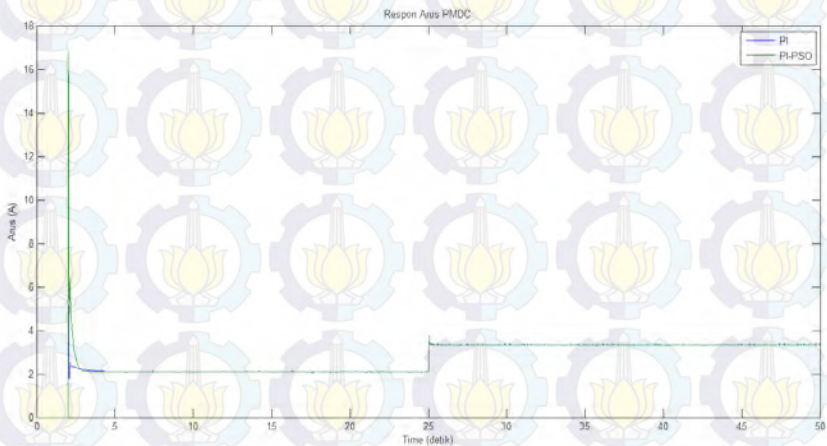
Dapat dilihat pada tabel 4.4 bahwa sistem dengan kontroler PI pada jangka waktu 25-50 detik tidak memiliki nilai *overshoot* dan memiliki nilai *settling time* sebesar 29,45 detik, sedangkan sistem dengan kontroler PI yang dioptimisasi dengan menggunakan PSO tidak memiliki nilai *overshoot* dan memiliki nilai *settling time* sebesar 26,05 detik. Oleh karena itu dapat disimpulkan bahwa pada periode 25-50 detik (pada saat sistem diberi gangguan beban) respon sistem dengan kontroler PI yang dioptimisasi dengan PSO lebih baik dibandingkan respon sistem dengan kontroler PI konvensional.

#### 4.2.1.2 Respon Arus ( $I_a$ ) Sistem Motor PMDC

Studi kasus yang pertama adalah mengamati respon arus sistem motor PMDC dengan gangguan torsi mekanik 1 Nm, sedangkan studi kasus yang kedua adalah mengamati respon arus sistem motor PMDC dengan gangguan torsi mekanik 2 Nm. Respon arus sistem motor PMDC dengan gangguan torsi mekanik 1 Nm ditunjukkan pada gambar 4.3. Pengamatan dilakukan pada detik ke-0 hingga detik ke-50 sehingga diperoleh respon arus sistem pada kondisi motor tidak diberi gangguan

torsi mekanik maupun kondisi pada saat motor diberi gangguan torsi mekanik.

Gambar 4.3 adalah gambar respon arus motor PMDC pada jangka waktu 0-50 detik. Motor PMDC mulai beroperasi pada detik ke-2. Garis biru pada gambar tersebut menunjukkan respon sistem yang dikontrol dengan kontroler PI dengan metode konvensional, sedangkan garis hijau pada gambar tersebut menunjukkan respon sistem yang dikontrol dengan kontroler PI yang dioptimisasi oleh algoritma PSO.



**Gambar 4.3** Respon Arus Motor PMDC dengan Gangguan Torsi Mekanik 1 Nm ( $\omega_{ref} = 100$  rad/sec)

Gambar 4.3 adalah gambar respon arus motor PMDC pada jangka waktu 0-50 detik. Motor PMDC mulai beroperasi pada detik ke-2. Pada kondisi *starting*, motor membutuhkan arus start yang tinggi untuk menghasilkan torsi elektrik yang besar untuk melawan torsi bebannya.

#### 4.2.1.2.1 Respon Arus Saat Sistem tidak diberi Gangguan Beban (TL = 0 Nm, $\omega_{ref} = 100$ rad/sec)

Pada saat motor tidak diberi gangguan beban nilai arus motor PMDC adalah 2,087 A. Perbandingan *overshoot* dan *settling time* respon arus motor PMDC pada jangka waktu 0-25 detik (tidak diberi gangguan beban) dapat dilihat pada tabel 4.5.



**Tabel 4.5** Perbandingan *overshoot* dan *settling time* respon arus motor PMDC pada 0-25 detik ( $\omega_{ref} = 100 \text{ rad/sec}$ )

Metode Kontrol	PI	PI PSO
<i>Overshoot</i> (A)	5,667	16,82
<i>Settling time</i> (det)	6,468	6,299

Dapat dilihat pada tabel 4.5 bahwa sistem dengan kontroler PI pada jangka waktu 0-25 detik memiliki *overshoot* sebesar 5,377 A dan nilai *settling time* sebesar 6,468 detik, sedangkan sistem dengan kontroler PI yang dioptimisasi dengan menggunakan PSO memiliki nilai *overshoot* sebesar 4,342 A dan memiliki nilai *settling time* sebesar 6,299 detik. Oleh karena itu dapat disimpulkan bahwa pada periode 0-25 detik (pada saat sistem tidak diberi gangguan beban) respon sistem dengan kontroler PI yang dioptimisasi dengan PSO lebih baik dibandingkan respon sistem dengan kontroler PI konvensional.

#### 4.2.1.2.2 Respon Arus Saat Sistem diberi Gangguan Beban ( $T_L = 1 \text{ Nm}$ , $\omega_{ref} = 100 \text{ rad/sec}$ )

Pada jangka waktu 25-50 detik terdapat gangguan berupa torsi mekanik sebesar 1 Nm. Ketika motor diberi gangguan berupa beban, maka kecepatannya akan turun. Untuk mencapai kecepatan referensinya kembali, maka motor butuh torsi yang besar untuk melawan torsi mekanik yang diberikan beban, maka dari itu arus motor menjadi naik agar torsi motor menjadi besar sehingga motor dapat mencapai kecepatan referensinya kembali. Oleh karena itu ketika motor diberi gangguan beban, maka nilai arus motor PMDC naik menjadi 3,338 A sehingga pada waktu 25-50 detik nilai arus motor menjadi 3,338 A. Perbandingan *overshoot* dan *settling time* respon arus motor PMDC pada jangka waktu 25-50 detik (diberi gangguan beban) dapat dilihat pada tabel 4.6.

**Tabel 4.6** Perbandingan *overshoot* dan *settling time* respon arus motor PMDC saat 25-50 detik ( $T_L = 1 \text{ Nm}$ ,  $\omega_{ref} = 100 \text{ rad/sec}$ )

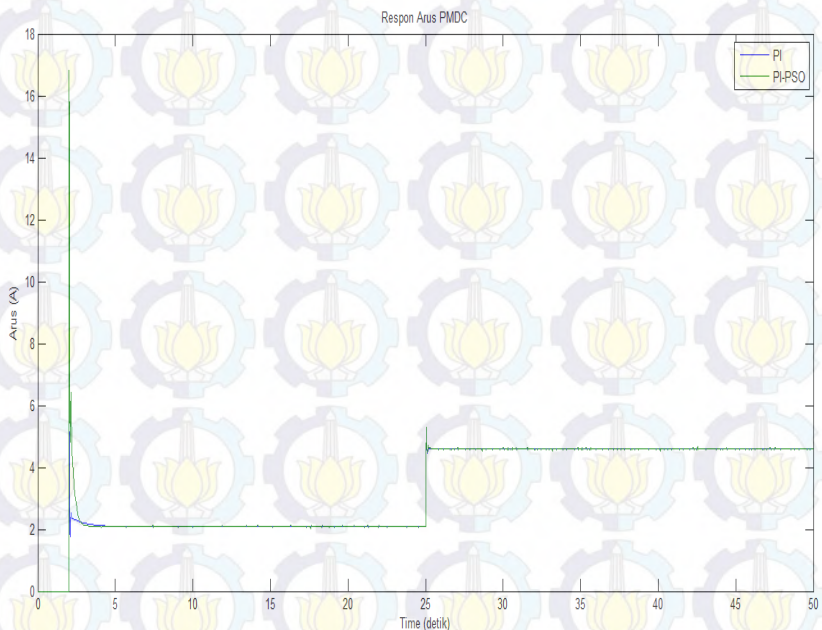
Metode Kontrol	PI	PI PSO
<i>Overshoot</i> (A)	3,616	3,738
<i>Settling time</i> (det)	25,96	25,43



Dapat dilihat pada tabel 4.6 bahwa sistem dengan kontroler PI pada jangka waktu 25-50 detik memiliki *overshoot* sebesar 3,616 A dan nilai *settling time* sebesar 25,96 detik, sedangkan sistem dengan kontroler PI yang dioptimisasi dengan menggunakan PSO memiliki nilai *overshoot* sebesar 3,738 A dan memiliki nilai *settling time* sebesar 25,43 detik. Oleh karena itu dapat disimpulkan bahwa pada periode 25-50 detik (pada saat sistem diberi gangguan beban) respon sistem dengan kontroler PI yang dioptimisasi dengan PSO lebih baik dibandingkan respon sistem dengan kontroler PI konvensional.

#### 4.2.1.1.2 Respon Arus Saat Sistem diberi Gangguan Beban ( $T_L = 2$ Nm, $\omega_{ref} = 100$ rad/sec)

Respon arus sistem motor PMDC dengan gangguan torsi mekanik 2 Nm ditunjukkan pada gambar 4.4.



**Gambar 4.4** Respon Arus Motor PMDC dengan Gangguan Torsi Mekanik 2 Nm ( $\omega_{ref} = 100$  rad/sec)

Pada jangka waktu 25-50 detik terdapat gangguan berupa torsi mekanik sebesar 2 Nm. Ketika motor diberi gangguan berupa beban, maka kecepatannya akan turun. Untuk mencapai kecepatan referensinya kembali, maka motor butuh torsi yang besar untuk melawan torsi mekanik yang diberikan beban, maka dari itu arus motor menjadi naik agar torsi motor menjadi besar sehingga motor dapat mencapai kecepatan referensinya kembali. Oleh karena itu ketika motor diberi gangguan beban, maka nilai arus motor PMDC naik menjadi 4,588 A sehingga pada waktu 25-50 detik nilai arus motor menjadi 4,588 A. Perbandingan *overshoot* dan *settling time* respon arus motor PMDC pada jangka waktu 25-50 detik (diberi gangguan beban) dapat dilihat pada tabel 4.7.

**Tabel 4.7** Perbandingan *overshoot* dan *settling time* respon arus motor PMDC saat 25-50 detik ( $T_L = 2 \text{ Nm}$ ,  $\omega_{ref} = 100 \text{ rad/sec}$ )

Metode Kontrol	PI	PI PSO
<i>Overshoot</i> (A)	5,07	5,283
<i>Settling time</i> (det)	26,48	26,05

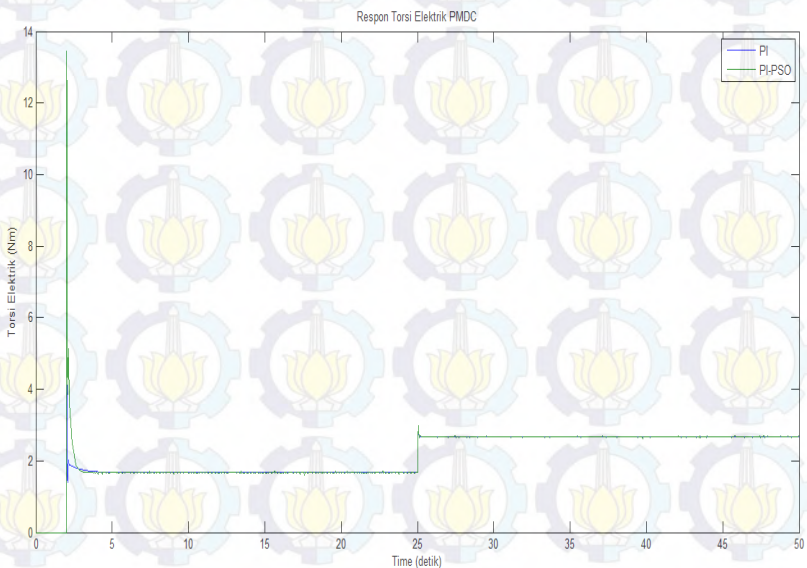
Dapat dilihat pada tabel 4.7 bahwa sistem dengan kontroler PI pada jangka waktu 25-50 detik memiliki *overshoot* sebesar 5,07 A dan nilai *settling time* sebesar 26,48 detik, sedangkan sistem dengan kontroler PI yang dioptimisasi dengan menggunakan PSO memiliki nilai *overshoot* sebesar 5,283 A dan memiliki nilai *settling time* sebesar 26,05 detik. Oleh karena itu dapat disimpulkan bahwa pada periode 25-50 detik (pada saat sistem diberi gangguan beban) respon sistem dengan kontroler PI yang dioptimisasi dengan PSO lebih baik dibandingkan respon sistem dengan kontroler PI konvensional.

#### 4.2.1.3 Respon Torsi Elektrik Motor ( $T_E$ ) Sistem Motor PMDC

Studi kasus yang pertama adalah mengamati respon torsi elektrik sistem motor PMDC dengan gangguan torsi mekanik 1 Nm, sedangkan studi kasus yang kedua adalah mengamati respon torsi elektrik sistem motor PMDC dengan gangguan torsi mekanik 2 Nm. Respon torsi elektrik sistem motor PMDC dengan gangguan torsi mekanik 1 Nm ditunjukkan pada gambar 4.5. Pengamatan dilakukan pada detik ke-0 hingga detik ke-50 sehingga diperoleh respon torsi

elektrik sistem pada kondisi motor tidak diberi gangguan torsi mekanik maupun kondisi pada saat motor diberi gangguan torsi mekanik.

Gambar 4.5 adalah gambar respon arus motor PMDC pada jangka waktu 0-50 detik saat motor diberi gangguan torsi mekanik. Motor PMDC mulai beroperasi pada detik ke-2. Garis biru pada gambar tersebut menunjukkan respon sistem yang dikontrol dengan kontroler PI dengan metode konvensional, sedangkan garis hijau pada gambar tersebut menunjukkan respon sistem yang dikontrol dengan kontroler PI yang dioptimisasi oleh algoritma PSO.



**Gambar 4.5** Respon Torsi Elektrik Motor PMDC dengan Gangguan Torsi Mekanik 1 Nm ( $\omega_{ref} = 100$  rad/sec)

#### 4.2.1.3.1 Respon Torsi Elektrik Saat Sistem tidak diberi Gangguan Beban ( $T_L = 0$ Nm, $\omega_{ref} = 100$ rad/sec)

Pada saat motor tidak diberi gangguan beban nilai torsi elektrik motor PMDC adalah 1,67 Nm. Perbandingan *overshoot* dan *settling time* respon torsi elektrik motor PMDC pada jangka waktu 0-25 detik (tidak diberi gangguan beban) dapat dilihat pada tabel 4.8.



**Tabel 4.8** Perbandingan *overshoot* dan *settling time* respon torsi elektrik motor PMDC saat 0-25 ( $\omega_{ref} = 100 \text{ rad/sec}$ )

Metode Kontrol	PI	PI PSO
<i>Overshoot</i> (rad/s)	4,534	13,46
<i>Settling time</i> (det)	6,468	6,299

Dapat dilihat pada tabel 4.8 bahwa sistem dengan kontroler PI pada jangka waktu 0-25 detik memiliki *overshoot* sebesar 4,534 A dan nilai *settling time* sebesar 6,468 detik, sedangkan sistem dengan kontroler PI yang dioptimisasi dengan menggunakan PSO memiliki nilai *overshoot* sebesar 13,46 A dan memiliki nilai *settling time* sebesar 6,299 detik. Oleh karena itu dapat disimpulkan bahwa pada periode 0-25 detik (pada saat sistem tidak diberi gangguan beban) respon sistem dengan kontroler PI yang dioptimisasi dengan PSO lebih baik dibandingkan respon sistem dengan kontroler PI konvensional.

#### 4.2.1.3.2 Respon Torsi Elektrik Motor Saat Sistem diberi Gangguan Beban ( $T_L = 1 \text{ Nm}$ , $\omega_{ref} = 100 \text{ rad/sec}$ )

Pada jangka waktu 25-50 detik terdapat gangguan berupa torsi mekanik sebesar 1 Nm. Ketika motor diberi gangguan berupa beban, maka kecepatannya akan turun. Untuk mencapai kecepatan referensinya kembali, maka motor butuh torsi yang besar untuk melawan torsi mekanik yang diberikan beban. Oleh karena itu ketika motor diberi gangguan beban, maka nilai torsi elektrik motor PMDC naik menjadi 2,67 Nm sehingga pada waktu 25-50 detik nilai torsi elektrik motor menjadi 2,67 Nm. Perbandingan *overshoot* dan *settling time* respon torsi elektrik motor PMDC pada jangka waktu 25-50 detik (diberi gangguan beban) dapat dilihat pada tabel 4.9.

**Tabel 4.9** Perbandingan *overshoot* dan *settling time* respon torsi elektrik motor PMDC saat 25-50 detik ( $T_L = 1 \text{ Nm}$ ,  $\omega_{ref} = 100 \text{ rad/sec}$ )

Metode Kontrol	PI	PI PSO
<i>Overshoot</i> (Nm)	2,893	2,99
<i>Settling time</i> (det)	26,44	25,96

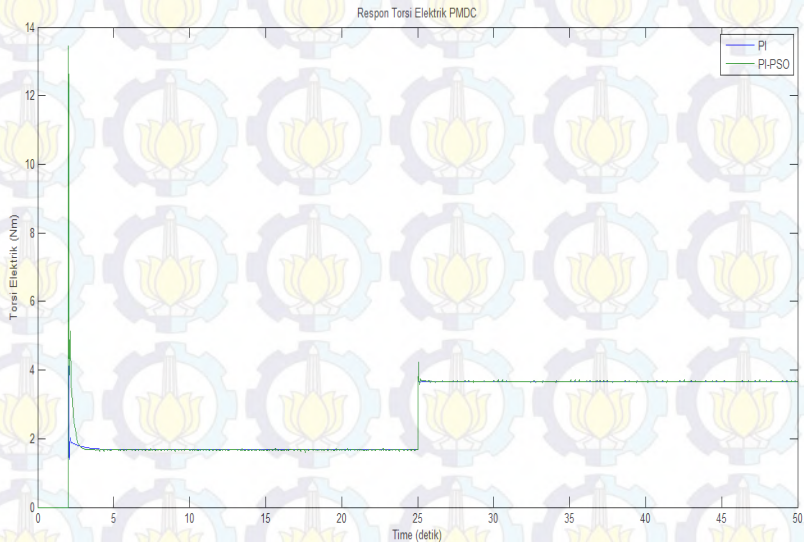
Dapat dilihat pada tabel 4.9 bahwa sistem dengan kontroler PI pada jangka waktu 25-50 detik memiliki *overshoot* sebesar 2,893 Nm



dan nilai *settling time* sebesar 26,44 detik, sedangkan sistem dengan kontroler PI yang dioptimisasi dengan menggunakan PSO memiliki nilai *overshoot* sebesar 2,99 Nm dan memiliki nilai *settling time* sebesar 25,96 detik. Oleh karena itu dapat disimpulkan bahwa pada periode 25-50 detik (pada saat sistem diberi gangguan beban) respon sistem dengan kontroler PI yang dioptimisasi dengan PSO lebih baik dibandingkan respon sistem dengan kontroler PI konvensional.

#### 4.2.1.3.3 Respon Torsi Elektrik Motor Saat Sistem diberi Gangguan Beban ( $T_L = 2 \text{ Nm}$ , $\omega_{ref} = 100 \text{ rad/sec}$ )

Respon torsi elektrik sistem motor PMDC dengan gangguan torsi mekanik 2 Nm ditunjukkan pada gambar 4.6.



**Gambar 4.6** Respon Torsi Elektrik Motor PMDC dengan Gangguan Torsi Mekanik 2 Nm ( $\omega_{ref} = 100 \text{ rad/sec}$ )

Pada jangka waktu 25-50 detik terdapat gangguan berupa torsi mekanik sebesar 2 Nm. Ketika motor diberi gangguan berupa beban, maka kecepatannya akan turun. Untuk mencapai kecepatan referensinya kembali, maka motor butuh torsi yang besar untuk melawan torsi

mekanik yang diberikan beban. Oleh karena itu ketika motor diberi gangguan beban, maka nilai torsi elektrik motor PMDC naik menjadi 3,67 Nm sehingga pada waktu 25-50 detik nilai torsi elektrik motor menjadi 3,67 Nm. Perbandingan *overshoot* dan *settling time* respon torsi elektrik motor PMDC pada jangka waktu 25-50 detik (diberi gangguan beban) dapat dilihat pada tabel 4.10.

**Tabel 4.10** Perbandingan *overshoot* dan *settling time* respon torsi elektrik motor PMDC saat 25-50 detik ( $T_L = 2 \text{ Nm}$ ,  $\omega_{ref} = 100 \text{ rad/sec}$ )

Metode Kontrol	PI	PI PSO
<i>Overshoot</i> (Nm)	4,056	4,227
<i>Settling time</i> (det)	27,08	26,84

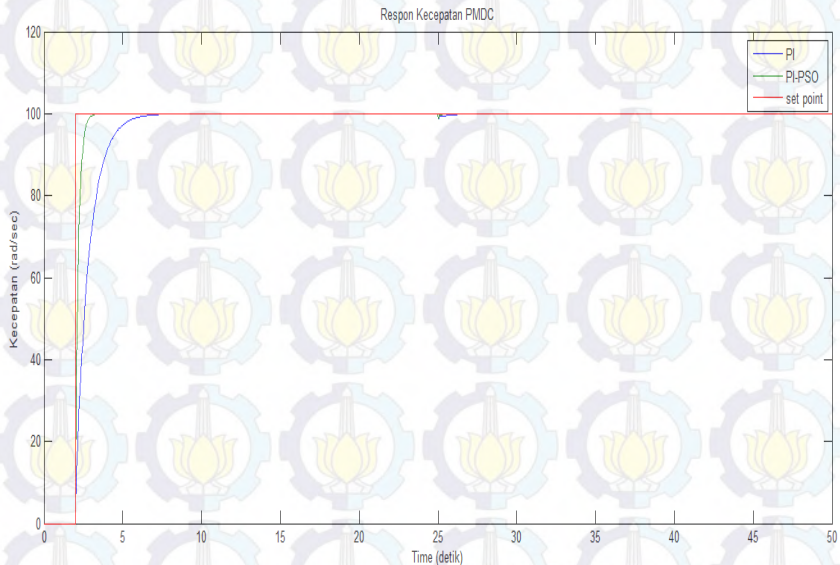
Dapat dilihat pada tabel 4.10 bahwa sistem dengan kontroler PI pada jangka waktu 25-50 detik memiliki *overshoot* sebesar 4,056 Nm dan nilai *settling time* sebesar 27,08 detik, sedangkan sistem dengan kontroler PI yang dioptimisasi dengan menggunakan PSO memiliki nilai *overshoot* sebesar 4,227 Nm dan memiliki nilai *settling time* sebesar 26,84 detik. Oleh karena itu dapat disimpulkan bahwa pada periode 25-50 detik (pada saat sistem diberi gangguan beban) respon sistem dengan kontroler PI yang dioptimisasi dengan PSO lebih baik dibandingkan respon sistem dengan kontroler PI konvensional.

#### 4.2.1.4 Respon Kecepatan ( $\omega_M$ ) Sistem Motor PMDC

Studi kasus yang pertama adalah mengamati respon kecepatan sistem motor PMDC dengan gangguan torsi mekanik 1 Nm, sedangkan studi kasus yang kedua adalah mengamati respon kecepatan sistem motor PMDC dengan gangguan torsi mekanik 2 Nm. Respon kecepatan sistem motor PMDC dengan gangguan torsi mekanik 1 Nm ditunjukkan pada gambar 4.7. Pengamatan dilakukan pada detik ke-0 hingga detik ke-50 sehingga diperoleh respon kecepatan sistem pada kondisi motor tidak diberi gangguan torsi mekanik maupun kondisi pada saat motor diberi gangguan torsi mekanik.

Gambar 4.7 adalah gambar respon arus motor PMDC pada jangka waktu 0-50 detik. Motor PMDC mulai beroperasi pada detik ke-2. Garis biru pada gambar tersebut menunjukkan respon sistem yang dikontrol dengan kontroler PI dengan metode konvensional, sedangkan

garis hijau pada gambar tersebut menunjukkan respon sistem yang dikontrol dengan kontroler PI yang dioptimisasi oleh algoritma PSO.



**Gambar 4.7** Respon Kecepatan Motor PMDC dengan Gangguan Torsi Mekanik 1 Nm ( $\omega_{ref} = 100$  rad/sec)

#### 4.2.1.4.1 Respon Kecepatan Saat Sistem tidak diberi Gangguan Beban ( $T_L = 0$ Nm, $\omega_{ref} = 100$ rad/sec)

Perbandingan *overshoot* dan *settling time* respon kecepatan motor PMDC pada jangka waktu 0-25 detik (tidak diberi gangguan beban) dapat dilihat pada tabel 4.11.

**Tabel 4.11** Perbandingan *overshoot* dan *settling time* respon kecepatan motor PMDC saat 0-25 ( $\omega_{ref} = 100$  rad/sec)

Metode Kontrol	PI	PI PSO
<i>Overshoot</i> (rad/s)	0	0
<i>Settling time</i> (det)	10,41	3,847

Dapat dilihat pada tabel 4.11 bahwa sistem dengan kontroler PI pada jangka waktu 0-25 detik memiliki nilai *settling time* sebesar 10,41



detik dan tidak memiliki nilai *overshoot*, sedangkan sistem dengan kontroler PI yang dioptimisasi dengan menggunakan PSO memiliki nilai *settling time* sebesar 3,847 detik dan juga tidak memiliki nilai *overshoot*. Oleh karena itu dapat disimpulkan bahwa pada periode 0-25 detik (pada saat sistem tidak diberi gangguan beban) respon sistem dengan kontroler PI yang dioptimisasi dengan PSO lebih baik dibandingkan respon sistem dengan kontroler PI konvensional karena respon sistem kembali ke kondisi *steady state* lebih cepat.

#### 4.2.1.4.2 Respon Kecepatan Saat Sistem diberi Gangguan Beban ( $T_L = 1 \text{ Nm}$ , $\omega_{ref} = 100 \text{ rad/sec}$ )

Pada jangka waktu 25-50 detik terdapat gangguan berupa torsi mekanik sebesar 5 Nm. Ketika motor diberi gangguan berupa beban, maka kecepatannya akan turun. Dari respon sistem sebelumnya dapat dilihat bahwa tegangan motor akan naik agar motor mencapai kecepatan referensinya kembali. Selain itu, pada saat kecepatan motor turun karena motor dibebani maka torsi dan arus motor akan naik agar torsi elektrik pada motor mampu melawan torsi mekanik yang diberikan beban sehingga dengan kondisi yang demikian motor dapat mencapai kecepatan referensinya kembali. Perbandingan *overshoot* dan *settling time* respon kecepatan motor PMDC pada jangka waktu 25-50 detik (diberi gangguan beban) dapat dilihat pada tabel 4.12.

**Tabel 4.12** Perbandingan *overshoot* dan *settling time* respon kecepatan motor PMDC saat 25-50 detik ( $T_L = 1 \text{ Nm}$ ,  $\omega_{ref} = 100 \text{ rad/sec}$ )

Metode Kontrol	PI	PI PSO
<i>Overshoot</i> (rad/s)	98,76	98,85
<i>Settling time</i> (det)	29,47	26,11

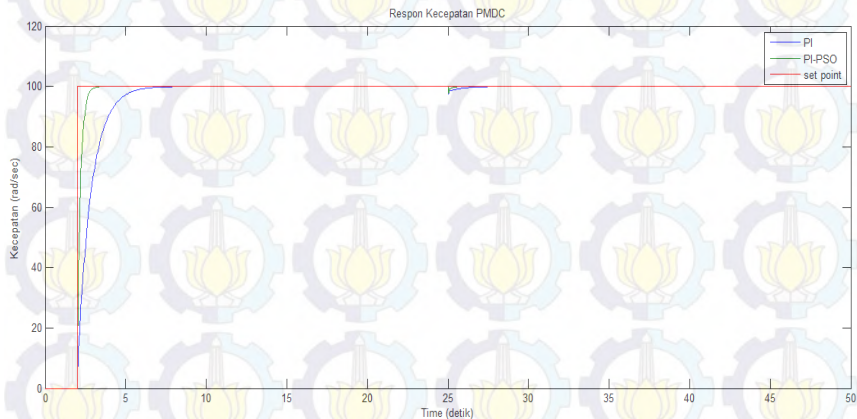
Dapat dilihat pada tabel 4.12 bahwa sistem dengan kontroler PI pada jangka waktu 25-50 detik memiliki nilai *overshoot* sebesar 98,76 rad/sec dan memiliki nilai *settling time* sebesar 29,47 detik, sedangkan sistem dengan kontroler PI yang dioptimisasi dengan menggunakan PSO memiliki nilai *overshoot* sebesar 98,85 rad/sec dan memiliki nilai *settling time* sebesar 26,11 detik. Dari tabel tersebut dapat diketahui bahwa pada saat sistem diberi gangguan beban, sistem dengan kontroler PI konvensional mengalami perlambatan kecepatan menjadi 98,76



rad/sec sedangkan sistem dengan kontroler PI yang dioptimisasi dengan menggunakan PSO mengalami perlambatan kecepatan menjadi 98,85 rad/sec. Itu artinya respon kecepatan motor PMDC yang dikontrol oleh kontroler PI konvensional lebih lambat dibandingkan dengan respon kecepatan motor PMDC yang dikontrol oleh kontroler PI yang dioptimisasi dengan PSO ketika motor tersebut diberi beban. Oleh karena itu dapat disimpulkan bahwa pada periode 25-50 detik (pada saat sistem diberi gangguan beban) respon sistem dengan kontroler PI yang dioptimisasi dengan PSO lebih baik dibandingkan respon sistem dengan kontroler PI konvensional karena respon sistem kembali ke kondisi *steady state* lebih cepat dan perlambatan sistem ketika diberi beban lebih kecil.

#### 4.2.1.4.3 Respon Kecepatan Saat Sistem diberi Gangguan Beban ( $T_L = 2 \text{ Nm}$ , $\omega_{ref} = 100 \text{ rad/sec}$ )

Respon kecepatan sistem motor PMDC dengan gangguan torsi mekanik 2 Nm ditunjukkan pada gambar 4.8.



**Gambar 4.8** Respon Kecepatan Motor PMDC dengan Gangguan Torsi Mekanik 2 Nm ( $\omega_{ref} = 100 \text{ rad/sec}$ )

Pada jangka waktu 25-50 detik terdapat gangguan berupa torsi mekanik sebesar 2 Nm. Ketika motor diberi gangguan berupa beban, maka kecepatannya akan turun. Dari respon sistem sebelumnya dapat dilihat bahwa tegangan motor akan naik agar motor mencapai kecepatan

referensinya kembali. Selain itu, pada saat kecepatan motor turun karena motor dibebani maka torsi dan arus motor akan naik agar torsi elektrik pada motor mampu melawan torsi mekanik yang diberikan beban sehingga dengan kondisi yang demikian motor dapat mencapai kecepatan referensinya kembali. Perbandingan *overshoot* dan *settling time* respon kecepatan motor PMDC pada jangka waktu 25-50 detik (diberi gangguan beban) dapat dilihat pada tabel 4.13.

**Tabel 4.13** Perbandingan *overshoot* dan *settling time* respon kecepatan motor PMDC saat 25-50 detik ( $T_L = 2 \text{ Nm}$ ,  $\omega_{ref} = 100 \text{ rad/sec}$ )

Metode Kontrol	PI	PI PSO
<i>Overshoot</i> (rad/s)	97,59	97,71
<i>Settling time</i> (det)	30,04	26,17

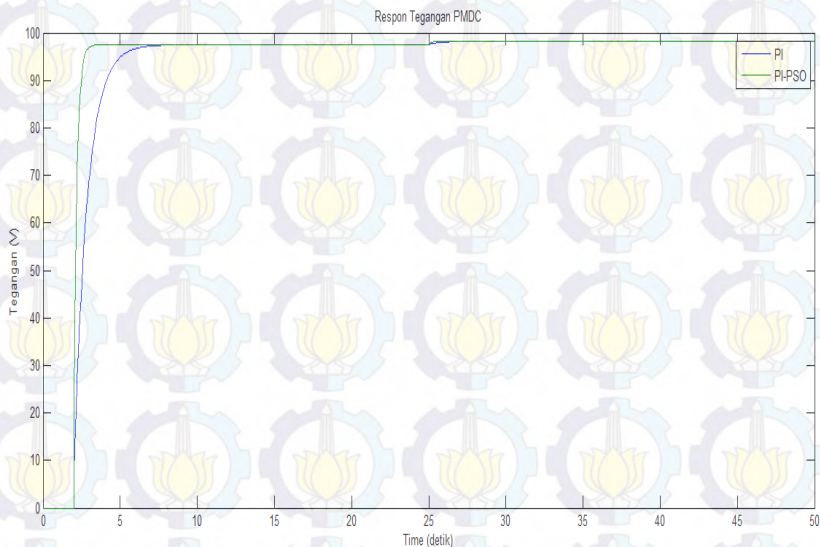
Dapat dilihat pada tabel 4.13 bahwa sistem dengan kontroler PI pada jangka waktu 25-50 detik memiliki nilai *overshoot* sebesar 97,59 rad/sec dan memiliki nilai *settling time* sebesar 30,04 detik, sedangkan sistem dengan kontroler PI yang dioptimisasi dengan menggunakan PSO memiliki nilai *overshoot* sebesar 97,71 rad/sec dan memiliki nilai *settling time* sebesar 26,17 detik. Dari tabel tersebut dapat diketahui bahwa pada saat sistem diberi gangguan beban, sistem dengan kontroler PI konvensional mengalami perlambatan kecepatan menjadi 97,59 rad/sec sedangkan sistem dengan kontroler PI yang dioptimisasi dengan menggunakan PSO mengalami perlambatan kecepatan menjadi 97,71 rad/sec. Itu artinya respon kecepatan motor PMDC yang dikontrol oleh kontroler PI konvensional lebih lambat dibandingkan dengan respon kecepatan motor PMDC yang dikontrol oleh kontroler PI yang dioptimisasi dengan PSO ketika motor tersebut diberi beban. Oleh karena itu dapat disimpulkan bahwa pada periode 25-50 detik (pada saat sistem diberi gangguan beban) respon sistem dengan kontroler PI yang dioptimisasi dengan PSO lebih baik dibandingkan respon sistem dengan kontroler PI konvensional karena respon sistem kembali ke kondisi *steady state* lebih cepat dan perlambatan sistem ketika diberi beban lebih kecil.

#### 4.2.2 Respon Sistem dengan Kecepatan Referensi 120 rad/sec

##### 4.2.2.1 Respon Tegangan (VT) Sistem Motor PMDC

Studi kasus yang pertama adalah mengamati respon tegangan sistem motor PMDC dengan gangguan torsi mekanik 1 Nm, sedangkan studi kasus yang kedua adalah mengamati respon tegangan sistem motor PMDC dengan gangguan torsi mekanik 2 Nm. Respon tegangan sistem motor PMDC dengan gangguan torsi mekanik 5 Nm ditunjukkan pada gambar 4.9. Pengamatan dilakukan pada detik ke-0 hingga detik ke-50 sehingga diperoleh respon tegangan pada sistem pada kondisi motor tidak diberi gangguan torsi mekanik maupun kondisi pada saat motor diberi gangguan torsi mekanik.

Gambar 4.9 adalah gambar respon tegangan motor PMDC pada jangka waktu 0-50 detik. Motor PMDC mulai beroperasi pada detik ke-2. Garis biru pada gambar tersebut menunjukkan respon sistem yang dikontrol dengan kontroler PI dengan metode konvensional, sedangkan garis hijau pada gambar tersebut menunjukkan respon sistem yang dikontrol dengan kontroler PI yang dioptimisasi oleh algoritma PSO.



**Gambar 4.9** Respon Tegangan Motor PMDC dengan Gangguan Torsi Mekanik 1 Nm ( $\omega_{ref} = 120$  rad/sec)

#### **4.2.2.1.1 Respon Tegangan Motor Saat Sistem tidak diberi Gangguan Beban ( $T_L = 0$ Nm, $\omega_{ref} = 120$ rad/sec)**



Pada saat motor tidak diberi gangguan beban nilai tegangan motor PMDC adalah 97,5 V. Perbandingan *overshoot* dan *settling time* respon tegangan motor PMDC pada jangka waktu 0-25 detik (tidak diberi gangguan beban) dapat dilihat pada tabel 4.14.

**Tabel 4.14** Perbandingan *overshoot* dan *settling time* respon tegangan motor PMDC pada 0-25 detik ( $\omega_{ref} = 120 \text{ rad/sec}$ )

Metode Kontrol	PI	PI PSO
<i>Overshoot</i> (V)	0	0
<i>Settling time</i> (det)	9,967	3,847

Dapat dilihat pada tabel 4.8 bahwa sistem dengan kontroler PI pada jangka waktu 0-25 detik tidak memiliki nilai *overshoot* dan memiliki nilai *settling time* sebesar 9,967 detik, sedangkan sistem dengan kontroler PI yang dioptimisasi dengan menggunakan PSO tidak memiliki nilai *overshoot* dan memiliki nilai *settling time* sebesar 3,847 detik. Oleh karena itu dapat disimpulkan bahwa pada periode 0-25 detik (pada saat sistem tidak diberi gangguan beban) respon sistem dengan kontroler PI yang dioptimisasi dengan PSO lebih baik dibandingkan respon sistem dengan kontroler PI konvensional karena respon sistem kembali ke kondisi *steady state* lebih cepat.

#### 4.2.2.1.2 Respon Tegangan Motor Saat Sistem diberi Gangguan Beban ( $T_L = 1 \text{ Nm}$ , $\omega_{ref} = 120 \text{ rad/sec}$ )

Pada jangka waktu 25-50 detik terdapat gangguan berupa torsi mekanik sebesar 1 Nm. Ketika motor diberi gangguan berupa beban, maka kecepatannya akan turun. Oleh karena itu tegangan input pada motor harus dinaikkan agar kecepatan motor dapat kembali ke kecepatan yang diinginkan karena salah satu cara dalam pengaturan kecepatan motor PMDC adalah dengan mengontrol tegangan inputnya. Oleh karena itu saat motor diberi beban torsi mekanik 1 Nm nilai tegangannya naik menjadi 98,25 V sehingga pada waktu 25-50 detik nilai tegangan motor menjadi 98,25 V. Perbandingan *overshoot* dan *settling time* respon tegangan motor PMDC pada jangka waktu 25-50 detik (diberi gangguan beban) dapat dilihat pada tabel 4.15.



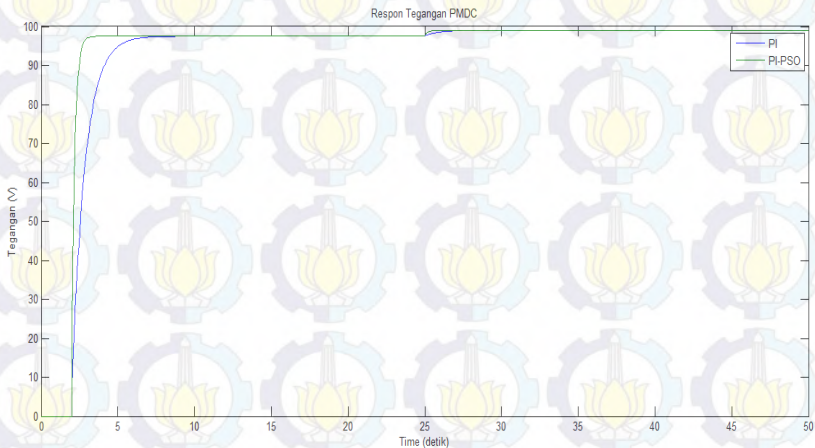
**Tabel 4.15** Perbandingan *overshoot* dan *settling time* respon tegangan motor PMDC saat 25-50 detik ( $T_L = 1 \text{ Nm}$ ,  $\omega_{\text{ref}} = 120 \text{ rad/sec}$ )

Metode Kontrol	PI	PI PSO
<i>Overshoot</i> (V)	0	0
<i>Settling time</i> (det)	28,81	25,9

Dapat dilihat pada tabel 4.9 bahwa sistem dengan kontroler PI pada jangka waktu 25-50 detik tidak memiliki nilai *overshoot* dan memiliki nilai *settling time* sebesar 28,81 detik, sedangkan sistem dengan kontroler PI yang dioptimisasi dengan menggunakan PSO tidak memiliki nilai *overshoot* dan memiliki nilai *settling time* sebesar 25,9 detik. Oleh karena itu dapat disimpulkan bahwa pada periode 25-50 detik (pada saat sistem diberi gangguan beban) respon sistem dengan kontroler PI yang dioptimisasi dengan PSO lebih baik dibandingkan respon sistem dengan kontroler PI konvensional.

#### 4.2.2.1.3 Respon Tegangan Motor Saat Sistem diberi Gangguan Beban ( $T_L = 2 \text{ Nm}$ , $\omega_{\text{ref}} = 120 \text{ rad/sec}$ )

Respon tegangan sistem motor PMDC dengan gangguan torsi mekanik 2 Nm ditunjukkan pada gambar 4.10.



**Gambar 4.10** Respon Tegangan Motor PMDC dengan Gangguan Torsi Mekanik 2 Nm ( $\omega_{\text{ref}} = 120 \text{ rad/sec}$ )

Pada jangka waktu 25-50 detik terdapat gangguan berupa torsi mekanik sebesar 2 Nm. Ketika motor diberi gangguan berupa beban, maka kecepatannya akan turun. Oleh karena itu tegangan input pada motor harus dinaikkan agar kecepatan motor dapat kembali ke kecepatan yang diinginkan karena salah satu cara dalam pengaturan kecepatan motor PMDC adalah dengan mengontrol tegangan inputnya. Oleh karena itu saat motor diberi beban torsi mekanik 2 Nm nilai tegangannya naik menjadi 99 V sehingga pada waktu 25-50 detik nilai tegangan motor menjadi 99 V. Batas tegangan maksimal yang dibatasi oleh *signal limiter* adalah 103,35 V (6% dari tegangan 97,5 V saat  $\omega_{ref} = 120$  rad/sec). Perbandingan *overshoot* dan *settling time* respon tegangan motor PMDC pada jangka waktu 25-50 detik (diberi gangguan beban) dapat dilihat pada tabel 4.16.

**Tabel 4.16** Perbandingan *overshoot* dan *settling time* respon tegangan motor PMDC saat 25-50 detik ( $T_L = 2$  Nm,  $\omega_{ref} = 120$  rad/sec)

Metode Kontrol	PI	PI PSO
<i>Overshoot</i> (V)	0	0
<i>Settling time</i> (det)	29,38	26

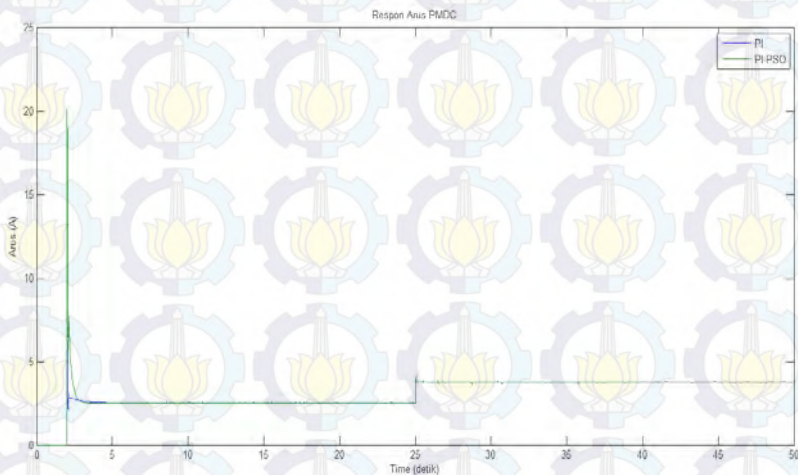
Dapat dilihat pada tabel 4.9 bahwa sistem dengan kontroler PI pada jangka waktu 25-50 detik tidak memiliki nilai *overshoot* dan memiliki nilai *settling time* sebesar 29,38 detik, sedangkan sistem dengan kontroler PI yang dioptimisasi dengan menggunakan PSO tidak memiliki nilai *overshoot* dan memiliki nilai *settling time* sebesar 26 detik. Oleh karena itu dapat disimpulkan bahwa pada periode 25-50 detik (pada saat sistem diberi gangguan beban) respon sistem dengan kontroler PI yang dioptimisasi dengan PSO lebih baik dibandingkan respon sistem dengan kontroler PI konvensional.

#### 4.2.2.2 Respon Arus ( $I_a$ ) Sistem Motor PMDC

Studi kasus yang pertama adalah mengamati respon arus sistem motor PMDC dengan gangguan torsi mekanik 1 Nm, sedangkan studi kasus yang kedua adalah mengamati respon arus sistem motor PMDC dengan gangguan torsi mekanik 2 Nm. Respon arus sistem motor PMDC dengan gangguan torsi mekanik 1 Nm ditunjukkan pada gambar 4.11. Pengamatan dilakukan pada detik ke-0 hingga detik ke-50 sehingga

diperoleh respon arus sistem pada kondisi motor tidak diberi gangguan torsi mekanik maupun kondisi pada saat motor diberi gangguan torsi mekanik.

Gambar 4.11 adalah gambar respon arus motor PMDC pada jangka waktu 0-50 detik. Motor PMDC mulai beroperasi pada detik ke-2. Garis biru pada gambar tersebut menunjukkan respon sistem yang dikontrol dengan kontroler PI dengan metode konvensional, sedangkan garis hijau pada gambar tersebut menunjukkan respon sistem yang dikontrol dengan kontroler PI yang dioptimisasi oleh algoritma PSO.



**Gambar 4.11** Respon Arus Motor PMDC dengan Gangguan Torsi Mekanik 1 Nm ( $\omega_{ref} = 120$  rad/sec)

Gambar 4.11 adalah gambar respon arus motor PMDC pada jangka waktu 0-50 detik. Motor PMDC mulai beroperasi pada detik ke-2. Pada kondisi *starting*, motor membutuhkan arus start yang tinggi untuk menghasilkan torsi elektrik yang besar untuk melawan torsi bebannya.

#### 4.2.2.2.1 Respon Arus Saat Sistem tidak diberi Gangguan Beban ( $T_L = 0$ Nm, $\omega_{ref} = 120$ rad/sec)

Pada saat motor tidak diberi gangguan beban nilai arus motor PMDC adalah 2,506 A. Perbandingan *overshoot* dan *settling time* respon



arus motor PMDC pada jangka waktu 0-25 detik (tidak diberi gangguan beban) dapat dilihat pada tabel 4.17.

**Tabel 4.17** Perbandingan *overshoot* dan *settling time* respon arus motor PMDC hingga pada 0-25 detik ( $\omega_{ref} = 120 \text{ rad/sec}$ )

Metode Kontrol	PI	PI PSO
<i>Overshoot</i> (A)	6,8	20,18
<i>Settling time</i> (det)	6,405	6,351

Dapat dilihat pada tabel 4.17 bahwa sistem dengan kontroler PI pada jangka waktu 0-25 detik memiliki *overshoot* sebesar 6,8 A dan nilai *settling time* sebesar 6,405 detik, sedangkan sistem dengan kontroler PI yang dioptimisasi dengan menggunakan PSO memiliki nilai *overshoot* sebesar 20,18 A dan memiliki nilai *settling time* sebesar 6,351 detik. Oleh karena itu dapat disimpulkan bahwa pada periode 0-25 detik (pada saat sistem tidak diberi gangguan beban) respon sistem dengan kontroler PI yang dioptimisasi dengan PSO lebih baik dibandingkan respon sistem dengan kontroler PI konvensional.

#### 4.2.2.2.2 Respon Arus Saat Sistem diberi Gangguan Beban ( $T_L = 1 \text{ Nm}$ , $\omega_{ref} = 120 \text{ rad/sec}$ )

Pada jangka waktu 25-50 detik terdapat gangguan berupa torsi mekanik sebesar 1 Nm. Ketika motor diberi gangguan berupa beban, maka kecepatannya akan turun. Untuk mencapai kecepatan referensinya kembali, maka motor butuh torsi yang besar untuk melawan torsi mekanik yang diberikan beban, maka dari itu arus motor menjadi naik agar torsi motor menjadi besar sehingga motor dapat mencapai kecepatan referensinya kembali. Oleh karena itu ketika motor diberi gangguan beban, maka nilai arus motor PMDC naik menjadi 3,757 A sehingga pada waktu 25-50 detik nilai arus motor menjadi 3,757 A. Perbandingan *overshoot* dan *settling time* respon arus motor PMDC pada jangka waktu 25-50 detik (diberi gangguan beban) dapat dilihat pada tabel 4.18.

**Tabel 4.18** Perbandingan *overshoot* dan *settling time* respon arus motor PMDC saat 25-50 detik ( $T_L = 1 \text{ Nm}$ ,  $\omega_{ref} = 120 \text{ rad/sec}$ )

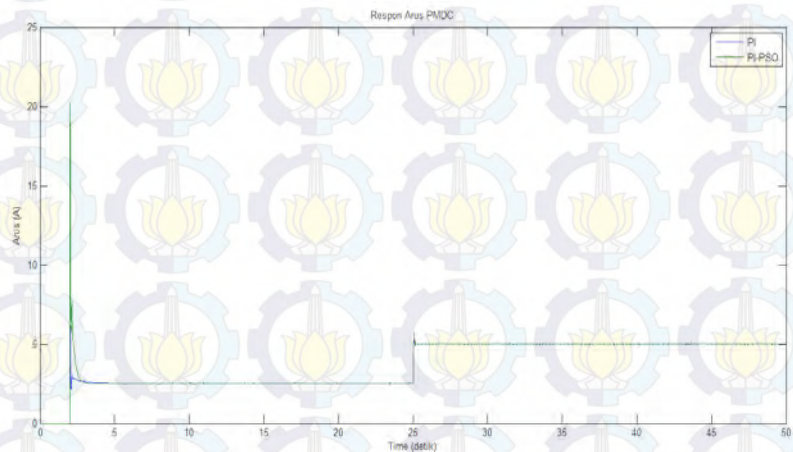
Metode Kontrol	PI	PI PSO
<i>Overshoot</i> (A)	4,039	4,152
<i>Settling time</i> (det)	26,15	25,8



Dapat dilihat pada tabel 4.18 bahwa sistem dengan kontroler PI pada jangka waktu 25-50 detik memiliki *overshoot* sebesar 4,039 A dan nilai *settling time* sebesar 26,15 detik, sedangkan sistem dengan kontroler PI yang dioptimisasi dengan menggunakan PSO memiliki nilai *overshoot* sebesar 4,152 A dan memiliki nilai *settling time* sebesar 25,8 detik. Oleh karena itu dapat disimpulkan bahwa pada periode 25-50 detik (pada saat sistem diberi gangguan beban) respon sistem dengan kontroler PI yang dioptimisasi dengan PSO lebih baik dibandingkan respon sistem dengan kontroler PI konvensional.

#### 4.2.2.2.3 Respon Arus Saat Sistem diberi Gangguan Beban ( $T_L = 2$ Nm, $\omega_{ref} = 120$ rad/sec)

Respon arus sistem motor PMDC dengan gangguan torsi mekanik 2 Nm ditunjukkan pada gambar 4.12.



**Gambar 4.12** Respon Arus Motor PMDC dengan Gangguan Torsi Mekanik 2 Nm ( $\omega_{ref} = 120$  rad/sec)

Pada jangka waktu 25-50 detik terdapat gangguan berupa torsi mekanik sebesar 2 Nm. Ketika motor diberi gangguan berupa beban, maka kecepatannya akan turun. Untuk mencapai kecepatan referensinya kembali, maka motor butuh torsi yang besar untuk melawan torsi mekanik yang diberikan beban, maka dari itu arus motor menjadi naik agar torsi motor menjadi besar sehingga motor dapat mencapai

kecepatan referensinya kembali. Oleh karena itu ketika motor diberi gangguan beban, maka nilai arus motor PMDC naik menjadi 5,005 A sehingga pada waktu 25-50 detik nilai arus motor menjadi 5,005 A. Perbandingan *overshoot* dan *settling time* respon arus motor PMDC pada jangka waktu 25-50 detik (diberi gangguan beban) dapat dilihat pada tabel 4.19.

**Tabel 4.19** Perbandingan *overshoot* dan *settling time* respon arus motor PMDC saat 25-50 detik ( $T_L = 2 \text{ Nm}$ ,  $\omega_{ref} = 100 \text{ rad/sec}$ )

Metode Kontrol	PI	PI PSO
<i>Overshoot</i> (A)	5,479	5,739
<i>Settling time</i> (det)	27,44	26,11

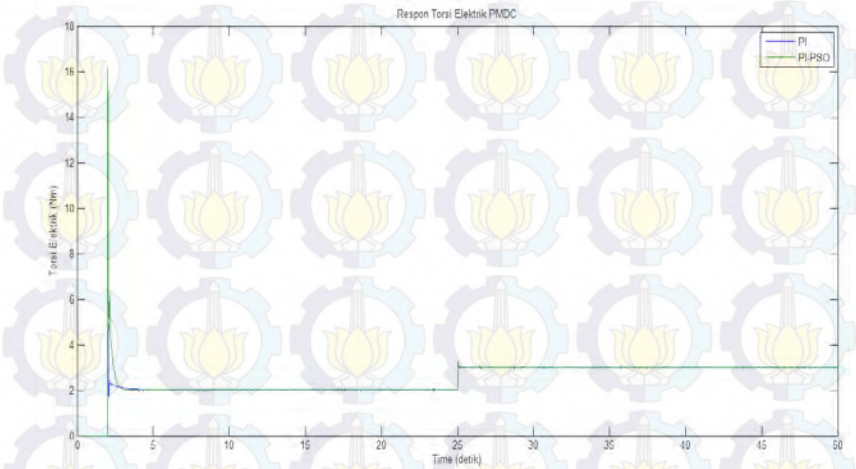
Dapat dilihat pada tabel 4.19 bahwa sistem dengan kontroler PI pada jangka waktu 25-50 detik memiliki *overshoot* sebesar 5,479 A dan nilai *settling time* sebesar 27,44 detik, sedangkan sistem dengan kontroler PI yang dioptimisasi dengan menggunakan PSO memiliki nilai *overshoot* sebesar 5,739 A dan memiliki nilai *settling time* sebesar 26,11 detik. Oleh karena itu dapat disimpulkan bahwa pada periode 25-50 detik (pada saat sistem diberi gangguan beban) respon sistem dengan kontroler PI yang dioptimisasi dengan PSO lebih baik dibandingkan respon sistem dengan kontroler PI konvensional.

#### 4.2.2.3 Respon Torsi Elektrik Motor ( $T_E$ ) Sistem Motor PMDC

Studi kasus yang pertama adalah mengamati respon torsi elektrik sistem motor PMDC dengan gangguan torsi mekanik 1 Nm, sedangkan studi kasus yang kedua adalah mengamati respon torsi elektrik sistem motor PMDC dengan gangguan torsi mekanik 2 Nm. Respon torsi elektrik sistem motor PMDC dengan gangguan torsi mekanik 1 Nm ditunjukkan pada gambar 4.13. Pengamatan dilakukan pada detik ke-0 hingga detik ke-50 sehingga diperoleh respon torsi elektrik sistem pada kondisi motor tidak diberi gangguan torsi mekanik maupun kondisi pada saat motor diberi gangguan torsi mekanik.

Gambar 4.13 adalah gambar respon arus motor PMDC pada jangka waktu 0-50 detik saat motor diberi gangguan torsi mekanik. Motor PMDC mulai beroperasi pada detik ke-2. Garis biru pada gambar tersebut menunjukkan respon sistem yang dikontrol dengan kontroler PI dengan metode konvensional, sedangkan garis hijau pada gambar

tersebut menunjukkan respon sistem yang dikontrol dengan kontroler PI yang dioptimisasi oleh algoritma PSO.



**Gambar 4.13**Respon Torsi Elektrik Motor PMDC dengan Gangguan Torsi Mekanik 1 Nm ( $\omega_{ref} = 120$  rad/sec)

**4.2.2.3.1**Respon Torsi Elektrik Saat Sistem tidak diberi Gangguan Beban ( $T_L = 0$  Nm,  $\omega_{ref} = 120$  rad/sec)

Pada saat motor tidak diberi gangguan beban nilai torsi elektrik motor PMDC adalah 2,004 Nm. Perbandingan *overshoot* dan *settling time* respon torsi elektrik motor PMDC pada jangka waktu 0-25 detik (tidak diberi gangguan beban) dapat dilihat pada tabel 4.20.

**Tabel 4.20** Perbandingan *overshoot* dan *settling time* respon torsi elektrik motor PMDC saat 0-25 detik ( $\omega_{ref} = 120$  rad/sec)

Metode Kontrol	PI	PI PSO
<i>Overshoot</i> (Nm)	5,44	16,15
<i>Settling time</i> (det)	6,469	6,299

Dapat dilihat pada tabel 4.20 bahwa sistem dengan kontroler PI pada jangka waktu 0-25 detik memiliki *overshoot* sebesar 5,44 Nm dan nilai *settling time* sebesar 6,469 detik, sedangkan sistem dengan kontroler PI yang dioptimisasi dengan menggunakan PSO memiliki nilai *overshoot* sebesar 16,15 Nm dan memiliki nilai *settling time* sebesar



6,299 detik. Oleh karena itu dapat disimpulkan bahwa pada periode 0-25 detik (pada saat sistem tidak diberi gangguan beban) respon sistem dengan kontroler PI yang dioptimisasi dengan PSO lebih baik dibandingkan respon sistem dengan kontroler PI konvensional.

#### 4.2.2.3.2 Respon Torsi Elektrik Motor Saat Sistem diberi Gangguan

**Beban ( $T_L = 1 \text{ Nm}$ ,  $\omega_{ref} = 120 \text{ rad/sec}$ )**

Pada jangka waktu 25-50 detik terdapat gangguan berupa torsi mekanik sebesar 1 Nm. Ketika motor diberi gangguan berupa beban, maka kecepatannya akan turun. Untuk mencapai kecepatan referensinya kembali, maka motor butuh torsi yang besar untuk melawan torsi mekanik yang diberikan beban. Oleh karena itu ketika motor diberi gangguan beban, maka nilai torsi elektrik motor PMDC naik menjadi 3,004 Nm sehingga pada waktu 25-50 detik nilai torsi elektrik motor menjadi 3,004 Nm. Perbandingan *overshoot* dan *settling time* respon torsi elektrik motor PMDC pada jangka waktu 25-50 detik (diberi gangguan beban) dapat dilihat pada tabel 4.21.

**Tabel 4.21** Perbandingan *overshoot* dan *settling time* respon torsi elektrik motor PMDC saat 25-50 detik ( $T_L = 1 \text{ Nm}$ ,  $\omega_{ref} = 120 \text{ rad/sec}$ )

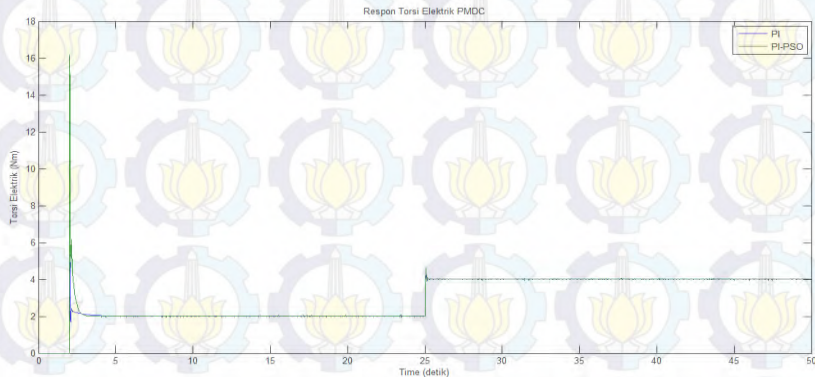
Metode Kontrol	PI	PI PSO
<i>Overshoot</i> (Nm)	3,231	3,322
<i>Settling time</i> (det)	26,3	25,42

Dapat dilihat pada tabel 4.21 bahwa sistem dengan kontroler PI pada jangka waktu 25-50 detik memiliki *overshoot* sebesar 3,231 Nm dan nilai *settling time* sebesar 26,3 detik, sedangkan sistem dengan kontroler PI yang dioptimisasi dengan menggunakan PSO memiliki nilai *overshoot* sebesar 3,322 Nm dan memiliki nilai *settling time* sebesar 25,42 detik. Oleh karena itu dapat disimpulkan bahwa pada periode 25-50 detik (pada saat sistem diberi gangguan beban) respon sistem dengan kontroler PI yang dioptimisasi dengan PSO lebih baik dibandingkan respon sistem dengan kontroler PI konvensional.

#### 4.2.2.3.3 Respon Torsi Elektrik Motor Saat Sistem diberi Gangguan

**Beban ( $T_L = 2 \text{ Nm}$ ,  $\omega_{ref} = 120 \text{ rad/sec}$ )**

Respon torsi elektrik sistem motor PMDC dengan gangguan torsi mekanik 2 Nm ditunjukkan pada gambar 4.14.



**Gambar 4.14** Respon Torsi Elektrik Motor PMDC dengan Gangguan Torsi Mekanik 2 Nm ( $\omega_{ref} = 120 \text{ rad/sec}$ )

Pada jangka waktu 25-50 detik terdapat gangguan berupa torsi mekanik sebesar 2 Nm. Ketika motor diberi gangguan berupa beban, maka kecepatannya akan turun. Untuk mencapai kecepatan referensinya kembali, maka motor butuh torsi yang besar untuk melawan torsi mekanik yang diberikan beban. Oleh karena itu ketika motor diberi gangguan beban, maka nilai torsi elektrik motor PMDC naik menjadi 4,004 Nm sehingga pada waktu 25-50 detik nilai torsi elektrik motor menjadi 4,004 Nm. Perbandingan *overshoot* dan *settling time* respon torsi elektrik motor PMDC pada jangka waktu 25-50 detik (diberi gangguan beban) dapat dilihat pada tabel 4.22.

**Tabel 4.22** Perbandingan *overshoot* dan *settling time* respon torsi elektrik motor PMDC saat 25-50 detik ( $T_L = 2 \text{ Nm}$ ,  $\omega_{ref} = 120 \text{ rad/sec}$ )

Metode Kontrol	PI	PI PSO
<i>Overshoot</i> (Nm)	4,384	4,591
<i>Settling time</i> (det)	27,44	26,11

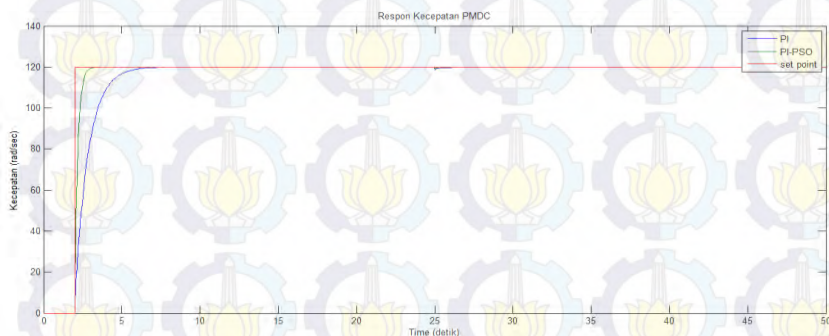
Dapat dilihat pada tabel 4.22 bahwa sistem dengan kontroler PI pada jangka waktu 25-50 detik memiliki *overshoot* sebesar 4,384 Nm

dan nilai *settling time* sebesar 27,44 detik, sedangkan sistem dengan kontroler PI yang dioptimisasi dengan menggunakan PSO memiliki nilai *overshoot* sebesar 4,592 Nm dan memiliki nilai *settling time* sebesar 26,11 detik. Oleh karena itu dapat disimpulkan bahwa pada periode 25-50 detik (pada saat sistem diberi gangguan beban) respon sistem dengan kontroler PI yang dioptimisasi dengan PSO lebih baik dibandingkan respon sistem dengan kontroler PI konvensional.

#### 4.2.2.4 Respon Kecepatan ( $\omega$ M) Sistem Motor PMDC

Studi kasus yang pertama adalah mengamati respon kecepatan sistem motor PMDC dengan gangguan torsi mekanik 1 Nm, sedangkan studi kasus yang kedua adalah mengamati respon kecepatan sistem motor PMDC dengan gangguan torsi mekanik 2 Nm. Respon kecepatan sistem motor PMDC dengan gangguan torsi mekanik 1 Nm ditunjukkan pada gambar 4.15. Pengamatan dilakukan pada detik ke-0 hingga detik ke-50 sehingga diperoleh respon kecepatan sistem pada kondisi motor tidak diberi gangguan torsi mekanik maupun kondisi pada saat motor diberi gangguan torsi mekanik.

Gambar 4.15 adalah gambar respon arus motor PMDC pada jangka waktu 0-50 detik. Motor PMDC mulai beroperasi pada detik ke-2. Garis biru pada gambar tersebut menunjukkan respon sistem yang dikontrol dengan kontroler PI dengan metode konvensional, sedangkan garis hijau pada gambar tersebut menunjukkan respon sistem yang dikontrol dengan kontroler PI yang dioptimisasi oleh algoritma PSO.



**Gambar 4.15** Respon Kecepatan Motor PMDC dengan Gangguan Torsi Mekanik 1 Nm ( $\omega_{ref} = 120$  rad/sec)



#### 4.2.2.4.1 Respon Kecepatan Saat Sistem tidak diberi Gangguan Beban ( $T_L = 0 \text{ Nm}$ , $\omega_{ref} = 120 \text{ rad/sec}$ )

Perbandingan *overshoot* dan *settling time* respon kecepatan motor PMDC pada jangka waktu 0-25 detik (tidak diberi gangguan beban) dapat dilihat pada tabel 4.23.

**Tabel 4.23** Perbandingan *overshoot* dan *settling time* respon kecepatan motor PMDC saat 0-25 ( $\omega_{ref} = 120 \text{ rad/sec}$ )

Metode Kontrol	PI	PI PSO
<i>Overshoot</i> (rad/s)	0	0
<i>Settling time</i> (det)	8,581	3,534

Dapat dilihat pada tabel 4.23 bahwa sistem dengan kontroler PI pada jangka waktu 0-25 detik memiliki nilai *settling time* sebesar 8,581 detik dan tidak memiliki nilai *overshoot*, sedangkan sistem dengan kontroler PI yang dioptimisasi dengan menggunakan PSO memiliki nilai *settling time* sebesar 3,534 detik dan juga tidak memiliki nilai *overshoot*. Oleh karena itu dapat disimpulkan bahwa pada periode 0-25 detik (pada saat sistem tidak diberi gangguan beban) respon sistem dengan kontroler PI yang dioptimisasi dengan PSO lebih baik dibandingkan respon sistem dengan kontroler PI konvensional karena respon sistem kembali ke kondisi *steady state* lebih cepat.

#### 4.2.2.4.2 Respon Kecepatan Saat Sistem diberi Gangguan Beban ( $T_L = 1 \text{ Nm}$ , $\omega_{ref} = 120 \text{ rad/sec}$ )

Pada jangka waktu 25-50 detik terdapat gangguan berupa torsi mekanik sebesar 1 Nm. Ketika motor diberi gangguan berupa beban, maka kecepatannya akan turun. Dari respon sistem sebelumnya dapat dilihat bahwa tegangan motor akan naik agar motor mencapai kecepatan referensinya kembali. Selain itu, pada saat kecepatan motor turun karena motor dibebani maka torsi dan arus motor akan naik agar torsi elektrik pada motor mampu melawan torsi mekanik yang diberikan beban sehingga dengan kondisi yang demikian motor dapat mencapai kecepatan referensinya kembali. Perbandingan *overshoot* dan *settling time* respon kecepatan motor PMDC pada jangka waktu 25-50 detik (diberi gangguan beban) dapat dilihat pada tabel 4.24.

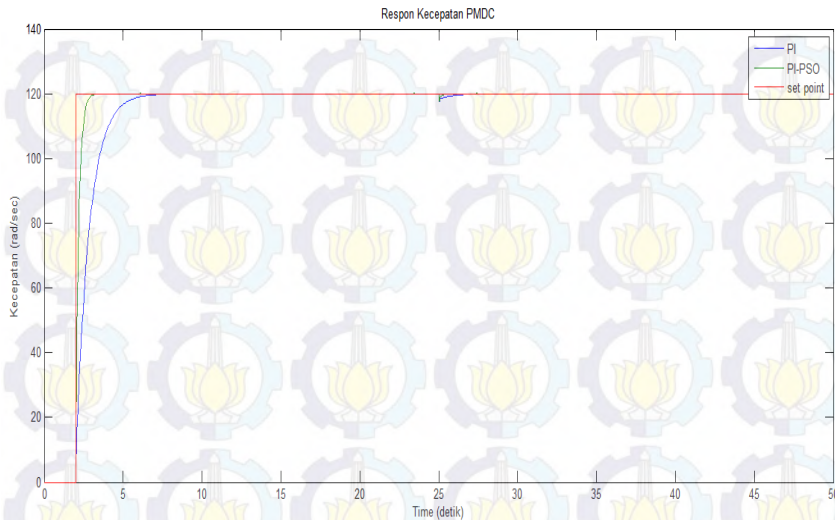
**Tabel 4.24** Perbandingan *overshoot* dan *settling time* respon kecepatan motor PMDC saat 25-50 detik ( $T_L = 1 \text{ Nm}$ ,  $\omega_{ref} = 120 \text{ rad/sec}$ )

Metode Kontrol	PI	PI PSO
<i>Overshoot</i> (rad/s)	118,8	118,9
<i>Settling time</i> (det)	27,46	25,61

Dapat dilihat pada tabel 4.24 bahwa sistem dengan kontroler PI pada jangka waktu 25-50 detik memiliki nilai *overshoot* sebesar 118,8 rad/sec dan memiliki nilai *settling time* sebesar 27,46 detik, sedangkan sistem dengan kontroler PI yang dioptimisasi dengan menggunakan PSO memiliki nilai *overshoot* sebesar 118,9 rad/sec dan memiliki nilai *settling time* sebesar 25,61 detik. Dari tabel tersebut dapat diketahui bahwa pada saat sistem diberi gangguan beban, sistem dengan kontroler PI konvensional mengalami perlambatan kecepatan menjadi 118,8 rad/sec sedangkan sistem dengan kontroler PI yang dioptimisasi dengan menggunakan PSO mengalami perlambatan kecepatan menjadi 118,9 rad/sec. Itu artinya respon kecepatan motor PMDC yang dikontrol oleh kontroler PI konvensional lebih lambat dibandingkan dengan respon kecepatan motor PMDC yang dikontrol oleh kontroler PI yang dioptimisasi dengan PSO ketika motor tersebut diberi beban. Oleh karena itu dapat disimpulkan bahwa pada periode 25-50 detik (pada saat sistem diberi gangguan beban) respon sistem dengan kontroler PI yang dioptimisasi dengan PSO lebih baik dibandingkan respon sistem dengan kontroler PI konvensional karena respon sistem kembali ke kondisi *steady state* lebih cepat dan perlambatan sistem ketika diberi beban lebih kecil.

#### 4.2.2.4.3 Respon Kecepatan Saat Sistem diberi Gangguan Beban ( $T_L = 2 \text{ Nm}$ , $\omega_{ref} = 120 \text{ rad/sec}$ )

Respon kecepatan sistem motor PMDC dengan gangguan torsi mekanik 2 Nm ditunjukkan pada gambar 4.16.



**Gambar 4.16** Respon Kecepatan Motor PMDC dengan Gangguan Torsi Mekanik 2 Nm ( $\omega_{ref} = 120$  rad/sec)

Pada jangka waktu 25-50 detik terdapat gangguan berupa torsi mekanik sebesar 2 Nm. Ketika motor diberi gangguan berupa beban, maka kecepatannya akan turun. Dari respon sistem sebelumnya dapat dilihat bahwa tegangan motor akan naik agar motor mencapai kecepatan referensinya kembali. Selain itu, pada saat kecepatan motor turun karena motor dibebani maka torsi dan arus motor akan naik agar torsi elektrik pada motor mampu melawan torsi mekanik yang diberikan beban sehingga dengan kondisi yang demikian motor dapat mencapai kecepatan referensinya kembali. Oleh karena itu tegangan input pada motor harus dinaikkan agar kecepatan motor dapat kembali ke kecepatan yang diinginkan karena salah satu cara dalam pengaturan kecepatan motor PMDC adalah dengan mengontrol tegangan inputnya. Perbandingan *overshoot* dan *settling time* respon kecepatan motor PMDC pada jangka waktu 25-50 detik (diberi gangguan beban) dapat dilihat pada tabel 4.25.



**Tabel 4.25** Perbandingan *overshoot* dan *settling time* respon kecepatan motor PMDC saat 25-50 detik (TL = 2 Nm,  $\omega_{ref}$  =120 rad/sec)

Metode Kontrol	PI	PI PSO
<i>Overshoot</i> (rad/s)	117,6	117,7
<i>Settling time</i> (det)	28,05	25,62

Dapat dilihat pada tabel 4.25 bahwa sistem dengan kontroler PI pada jangka waktu 25-50 detik memiliki nilai *overshoot* sebesar 117,6 rad/sec dan memiliki nilai *settling time* sebesar 28,05 detik, sedangkan sistem dengan kontroler PI yang dioptimisasi dengan menggunakan PSO memiliki nilai *overshoot* sebesar 117,7 rad/sec dan memiliki nilai *settling time* sebesar 25,62 detik. Dari tabel tersebut dapat diketahui bahwa pada saat sistem diberi gangguan beban, sistem dengan kontroler PI konvensional mengalami perlambatan kecepatan menjadi 117,6 rad/sec sedangkan sistem dengan kontroler PI yang dioptimisasi dengan menggunakan PSO mengalami perlambatan kecepatan menjadi 117,7 rad/sec. Itu artinya respon kecepatan motor PMDC yang dikontrol oleh kontroler PI konvensional lebih lambat dibandingkan dengan respon kecepatan motor PMDC yang dikontrol oleh kontroler PI yang dioptimisasi dengan PSO ketika motor tersebut diberi beban. Oleh karena itu dapat disimpulkan bahwa pada periode 25-50 detik (pada saat sistem diberi gangguan beban) respon sistem dengan kontroler PI yang dioptimisasi dengan PSO lebih baik dibandingkan respon sistem dengan kontroler PI konvensional karena respon sistem kembali ke kondisi *steady state* lebih cepat dan perlambatan sistem ketika diberi beban lebih kecil.

Dari hasil simulasi pada keempat respon sistem diatas dapat dilihat bahwa sistem yang menggunakan kontroler PI yang dioptimisasi dengan PSO memiliki respon yang paling baik. Hal ini menunjukkan bahwa dengan menggunakan PSO didapatkan koordinasi parameter PI yang optimal sehingga mampu meredam osilasi sistem yang lebih baik dibandingkan kontroler yang lain.

## **BAB 5**

### **PENUTUP**

#### **5.1 Kesimpulan**

Dari hasil simulasi diperoleh beberapa kesimpulan yaitu sebagai berikut :

1. Metode PSO dapat digunakan untuk menentukan koordinasi parameter PI yang optimal.
2. Kontroler PI yang optimal dapat diterapkan pada sistem motor PMDC untuk meredam osilasi respon sistem motor PMDC.
3. Penerapan PSO terhadap PI pada sistem motor PMDC dapat memperkecil *overshoot* respon sistem dan mempercepat *settling time* sistem.
4. Perubahan respon sistem akan terjadi ketika motor diberi gangguan beban.

#### **5.2 Saran**

Saran untuk penelitian berikutnya adalah :

1. Untuk mendapatkan koordinasi parameter PI yang optimal pada sistem motor PMDC dapat dilakukan dengan menggunakan komputasi cerdas yang lain untuk mendapatkan hasil yang lebih optimal.
2. Kontroler PI dapat diterapkan pada sistem yang lebih besar.

## DAFTAR PUSTAKA

- [1] Imam Robandi, "Modern Power System Control", Penerbit ANDI, Yogyakarta, 2009.
- [2] Imam Robandi, "Desain Sistem Tenaga Modern", Penerbit ANDI, Yogyakarta, 2006.
- [3] Soebagio, "Teori Umum Mesin Elektrik", Penerbit Srikandi, Surabaya, 2008.
- [4] Zuhul, "Dasar Tenaga Listrik", Penerbit ITB, Bandung, 1991.
- [5] Djiteng Marsudi, "Operasi Sistem Tenaga Listrik". 2006, Graha Ilmu, Yogyakarta. ISBN 978-756.
- [6] Robert Babuska, Stefano Stramigioli, "Matlab and Simulink for Modelling and Control", Delft University of Technology, 1999.
- [7] M.V. Deshpande, "Electric Motors Applications and Control", 1990, published by Y.P. Choopra for A. Wheeler & Co. Ltd., 23, L. B. Shastri Marg Allahabad-211 001.
- [8] Jimmie J. Cathey, "Electricmachines Analysis and Design Applying Matlab", University of Kentucky USA, 2001
- [9] Katsuhiko Ogata, "Teknik Kontrol Automatik", Penerbit Erlangga, Jakarta, 1989.
- [10] Iwan Setiawan, "Kontrol PID Untuk Proses Industri", Penerbit PT Elex Media Komputindo, Jakarta, 2008.
- [11] Sulasno, "Teknik Konversi Energi Listrik dan Sistem Pengaturan", Penerbit Graha Ilmu, Yogyakarta, 2009.
- [12] Uma maheswararao.Ch, Y.S.Kishore Babu, K.Amaresh "Sliding Mode Speed Control of a DC Motor", International Conference on Communication Systems and Network Technologies, 2011.
- [13] Ghazanfar Shahgolian, Pegah Shafaghi, "State Space Modeling and Eigenvalue Analysis of the Permanent Magnet DC Motor Drive System", IEEE press, Department of Electrical Engineering Islamic Azad University Iran , 2010.
- [14] Adel M. Sharaf, Adel A.A. El-Gammal, "A Novel Particle Swarm Optimization PSO Tuning Scheme for PMDC Motor Drives Controllers", IEEE press, University of Trinidad and Tobago, 2009.
- [15] Adel M. Sharaf, Adel A.A. El-Gammal, "An Integral Squared Error –ISE Optimal Parameters Tuning of Modified PID



- Controller for Industrial PMDC Motor Based on Particle Swarm Optimization –PSO”, IEEE press, University of Trinidad and Tobago, 2009.
- [16] Adel M. Sharaf, Adel A.A. El-Gammal, “Optimal Multi Incremental Self Regulating Speed Controller for Industrial PMDC Motor Drive Systems”, IEEE press, University of Trinidad and Tobago, 2009.
- [17] Asim H. Syed, M.A. Abido, “Differential Evolution based Intelligent Control for Speed Regulation of a PMDC Motor”, IEEE press, Greece, 2013.
- [18] Aashish Kumar Bohre, Dr. Ganga Agnihotri, Dr. Manisha Dubey, “Hybrid Butterfly Based Particle Swarm Optimization for Optimization Problems”, IEEE press, India, 2014 .
- [19] K.H. Ng, C.F. Yeong, E.L.M. Su, A.R. Husain, “Current Sensorless Control of a PMDC Motor using Kalman Filter and Cascaded PID Controller”, IEEE press, Universiti Teknologi Malaysia, 2012.
- [20] Yongbin Ma, Yongxin Liu, Cun Wang, “Design of Parameters Self-tuning Fuzzy PID Control for DC Motor”, IEEE press, Inner Mongolia University China, 2010
- [21] Otto Bucsu, Gabor Kawai, Istvan Kecskes, “Modeling the Neuro-Fuzzy Control with the Dynamic Model of the Permanent Magnet DC Motor”, 4<sup>th</sup> Serbian-Hungarian Joint Symposium on Intelligent Systems, 2006.
- [22] Ahmad A. Mahfouz, Mohammed M.K., Farhan A. Salem “Modeling, Simulation and Dynamics Analysis Issues of Electric Motor, for Mechatronics Applications, Using Different Approaches and Verification by MATLAB/Simulink”, I.J. Intelligent Systems and Applications, MECS, 2013.
- [23] Boumediène Allaoua, Brahim Gasbaoui and Brahim Mebarki, “Setting Up PID DC Motor Speed Control Alteration Parameters Using Particle Swarm Optimization Strategy”, Bechar University Algeria, Leonardo Electronic Journal of Practices and Technologies, 2009.
- [24] Haytham M. Fayek and I. Elamvazuthi, “Type-2 Fuzzy Logic PI (T2FLPI) Based DC Servomotor Control”, Journal of Applied Sciences Research ISSN, Universiti Teknologi PETRONAS Malaysia, 2012.

- [25] M.B.B. Sharifian, R.Rahnavard and H.Delavari, "Velocity Control of DC Motor Based Intelligent methods and Optimal Integral State Feedback Controller", International Journal of Computer Theory and Engineering, Iran, 2009.
- [26] Wendy Kurniawan Kautsar, "Koordinasi Kontroler PID dan Thyristor Controlled Phase Shifter (TCPS) Pada Load Frequency Control (LFC) Menggunakan Differential Evolution (DE)", Tugas Akhir Jurusan Teknik Elektro FTI-ITS, Surabaya, 2012.

## RIWAYAT HIDUP PENULIS



Penulis bernama lengkap Rodhi Kelvianto adalah anak pertama dari tiga bersaudara dari pasangan Bapak Sukartono dan Ibu Priyati Utami, yang lahir di Jakarta, 14 Maret 1993. Penulis memulai jenjang pendidikan sekolah dasar di SDI Al Azhar 6 Bekasi. Setelah lulus sekolah dasar tahun 2005, penulis melanjutkan sekolah ke SMP Negeri 92 Jakarta Timur dan lulus tahun 2008. Kemudian, penulis melanjutkan sekolah di SMA Negeri 31 Jakarta Timur. Setelah lulus dari SMAN 31 Jakarta

Timur tahun 2011, penulis melanjutkan studi S1 di Institut Teknologi Sepuluh Nopember (ITS) Surabaya jurusan Teknik Elektro dan mengambil konsentrasi bidang studi Teknik Sistem Tenaga. Selama menempuh studi S1, penulis aktif dalam berbagai organisasi pergerakan mahasiswa yaitu di Departemen PSDM HIMATEKTRO ITS, Kementerian Dagri BEM ITS, Kebijakan Publik KAMMI Sepuluh Nopember, serta yang terakhir adalah Ketua Mahkamah Mahasiswa ITS sebelum penulis lulus. Penulis juga pernah menjadi Ketua Umum Lembaga Dakwah Jurusan (LDJ) Kalam Elektro ITS dan anggota Elemen Pemuda Surabaya, yaitu organisasi pergerakan mahasiswa Surabaya yang dibentuk untuk mendukung Pemkot Surabaya dalam penutupan prostitusi Dolly tahun 2014. Penulis dapat dihubungi melalui :

Email	: <a href="mailto:rodhi.kelvianto@yahoo.com">rodhi.kelvianto@yahoo.com</a>
facebook	: Rodhi Kelvianto
Twitter	: @RodhiKelvianto1
Instagram	: @rodhikelvianto
Path	: Rodhi Kelvianto